



**Technische Universität Ilmenau**

Fakultät Informatik-Automatisierung

Fachgebiet Grafische Datenverarbeitung

# **Entwicklung und Implementation einer unscharfen Auswahl von Direktantrieben**

Diplomarbeit zur Erlangung des akademischen Grades [Diplominformatiker]

**Torsten Meß**

Betreuer: Jun.-Prof. Dr.-Ing. T. Ströhla, FG Entwurf Mechatronischer Antriebe  
Dipl.-Ing. T. Erbe, FG Feinwerktechnik  
Dipl.-Inf. U. Döring, FG Grafische Datenverarbeitung

Verantwortlicher Hochschullehrer:

Prof. Dr. sc. techn. B. Brüderlin, FG Grafische Datenverarbeitung

Die Diplomarbeit wurde am 27.05.2009 bei der Fakultät für Informatik und  
Automatisierung der Technischen Universität Ilmenau eingereicht.

Inventarisierungsnummer: 2009-02-27/030/IN00/2252

# Danksagung

An dieser Stelle möchte ich mich bei allen Personen bedanken, die mir bei der Durchführung der Interviews und den Fragebögen geholfen haben.

Prof. Brüderlin und Dipl. Inf. Döring möchte ich für die Übernahme der Betreuung meiner Diplomarbeit und den damit verbunden Aufwand ganz herzlich danken. Weiterhin möchte ich auch Herrn Plentz und Herrn Emam für die Unterstützung und gute Zusammenarbeit danken.

Einen grossen Dank möchte ich Prof. Dr.-Ing Ströhla und Dipl. Ing. Erbe aussprechen, die mich während der ganzen Zeit mit einer hervorragenden Betreuung unterstützt haben und mit nützlichen Hinweisen zur Seite standen. Besonders Dipl. Ing. Erbe möchte dafür danken, dass er sich für die vielen Konsultationen, dem Helfen bei den Interviews und den Fragebögen und dem mehrfachen kritischen Lesen und damit Verbessern der Diplomarbeit Zeit genommen hat.

Weiterhin möchte ich meiner Familie und meinen Freunden danken, die mich während meines Studiums und der Diplomarbeit unterstützt und immer an mich geglaubt haben.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Problemstellung . . . . .	1
1.2	Ziel der Arbeit . . . . .	2
1.3	Aufbau der Arbeit . . . . .	3
<b>2</b>	<b>Grundlagen</b>	<b>4</b>
2.1	Begriffsabgrenzung Akteur und Direktantrieb . . . . .	4
2.2	Funktion-Struktur-Speicher . . . . .	5
2.3	Usability . . . . .	6
2.4	Usability Engineering Lifecycle . . . . .	8
<b>3</b>	<b>Anforderungsanalyse</b>	<b>14</b>
3.1	Nutzerprofil . . . . .	14
3.2	Aufgabenanalyse . . . . .	15
3.2.1	Zielsetzung und Vorgehensweise . . . . .	15
3.2.2	Ergebnisse . . . . .	18
3.3	Ableiten von Usability-Zielen . . . . .	22
3.3.1	Zielsetzung . . . . .	22
3.3.2	Vorgehensweise . . . . .	24
3.3.3	Ergebnisse . . . . .	25
3.4	Plattformbeschränkungen . . . . .	25
3.5	Designprinzipien . . . . .	29

---

<b>4</b>	<b>Design</b>	<b>32</b>
4.1	Work-Reengineering . . . . .	33
4.2	Conceptual-Model-Design . . . . .	35
4.2.1	Zielsetzung . . . . .	35
4.2.2	Vorgehensweise . . . . .	35
4.2.3	Ergebnis . . . . .	37
4.3	Screen-Design und detailliertes UI-Design . . . . .	38
4.3.1	Suchinterface . . . . .	38
4.3.2	Ergebnisvisualisierung . . . . .	42
4.3.3	Ergebnistabelle . . . . .	60
4.4	Designergebnis . . . . .	63
<b>5</b>	<b>Entwicklung/Umsetzung</b>	<b>65</b>
5.1	Suchinterface . . . . .	68
5.2	Ergebnisvisualisierung . . . . .	69
5.3	Ergebnistabelle . . . . .	71
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>75</b>
6.1	Zusammenfassung . . . . .	75
6.2	Ausblick . . . . .	76
6.2.1	Evaluation . . . . .	76
6.2.2	Suchinterface . . . . .	78
6.2.3	Ergebnisvisualisierung . . . . .	78
6.2.4	Ergebnistabelle . . . . .	79
6.2.5	Sonstige Funktionen . . . . .	80
<b>A</b>	<b>Nutzerprofil-Fragebogen</b>	<b>85</b>
<b>B</b>	<b>Aufgabenanalyse-Fragebogen</b>	<b>90</b>

---

# Abbildungsverzeichnis

2.1	Struktur eines Antriebssystems (vgl. [KB91], [Zan07]) . . . . .	5
2.2	Grundkonzept des Funktion-Struktur-Speichers (vgl. [Han76], [Win08])	6
2.3	Überblick Usability Engineering Lifecycle [May99] . . . . .	10
3.1	Überblick Usability-Ziele . . . . .	23
3.2	Bildschirmauflösungs-Statistiken [w3s09] . . . . .	28
3.3	Browser-Statistiken [w3s08] . . . . .	28
4.1	Conceptual-Model-Design des User Interface (UI) . . . . .	36
4.2	Variante 1: Gleichzeitige Anzeige aller Parameter . . . . .	39
4.3	Variante 2: Anzeige ausschließlich der gesuchten Parameter . . . . .	39
4.4	Variante 2 mit Schieberegler . . . . .	41
4.5	Beispiel Parallel Coordinate Plot . . . . .	44
4.6	Beispiel Star Plot . . . . .	45
4.7	Beispiel Scatter Plot . . . . .	47
4.8	FilmFinder (vgl. [AS94]) . . . . .	48
4.9	Beispiel Bubble Chart . . . . .	49
4.10	Beispiel Clustered Bar Chart (a) und Stacked Bar Chart (b) . . . . .	50
4.11	Beispiel Histogramm . . . . .	52
4.12	Attribute Explorer . . . . .	53
4.13	Table Lens . . . . .	54
4.14	Bubble Chart als umgesetzte Visualisierung . . . . .	57
4.15	Zoom- und Scroll-Funktion in der Visualisierung . . . . .	59
4.16	Erweiterte Tabelle mit Balkendarstellung . . . . .	61

---

4.17 Normale Tabelle mit Zahlendarstellung . . . . .	61
4.18 Designergebnis . . . . .	64
5.1 Ablauf beim Besuchen einer PHP Seite . . . . .	66

# Tabellenverzeichnis

3.1	Überblick Nutzerprofile . . . . .	16
3.2	Überblick Usability-Ziele . . . . .	26
5.1	Vergleich Visualisierungs-Frameworks . . . . .	70
5.2	Vergleich von Frameworks und JavaScripts für die Ergebnistabelle . . .	72

# Abkürzungsverzeichnis

CSS Cascading Style Sheets

DOM Document Object Model

FSS Funktion-Struktur-Speicher

GUI Graphical User Interface

HTML Hypertext Markup Language

IE Internet Explorer

KEP Konstruktiver Entwicklungsprozess

MAMP Mac Apache MySQL PHP

PHP PHP: Hypertext Preprocessor

SQL Structured Query Language

UEL Usability Engineering Lifecycle

UI User Interface

URL Uniform Resource Locator

W3C World Wide Web Consortium

XHTML Extensible Hypertext Markup Language

XML Extensible Markup Language



# Kapitel 1

## Einleitung

### 1.1 Problemstellung

Ein essentieller Bestandteil moderner mechatronischer Systeme sind Aktoren. Ihre Aufgabe besteht darin, definierte Bewegungs-, Kraft- bzw. Momentenverläufe in ein oder mehr Freiheiten zu erzeugen. Um Zeit bei der Entwicklung mechatronischer Systeme zu sparen und somit die Kosten zu reduzieren, ist eine systematische Vorgehensweise beim Konstruktionsprozess von enormer Bedeutung. Bei der Entwicklung mechatronischer Systeme besteht die Schwierigkeit, für eine gegebene Problemstellung den optimalen Antrieb zu wählen [Zan07]. Die große Anzahl von Aktoren und Antriebssystemen, die sich im Laufe der Jahre entwickelt haben, erschwert es dem Ingenieur, eine geeignete Antriebsstruktur für seine Bewegungsaufgabe zu finden. Die Antriebssysteme können sich sowohl durch den physikalischen Effekt, auf dem sie beruhen, als auch strukturell unterscheiden. Der Ingenieur hat eine Vielzahl von Kriterien und Eigenschaften bei der Entwicklung mechatronischer Systeme zu bedenken. Diese können sich teilweise widersprechen und machen einen systematischen Vergleich oft unmöglich [Win08].

Des Weiteren sind Ingenieure durch ihre Erfahrungen oft dazu geneigt, eine Lösung in Bereichen zu suchen, mit denen sie schon zu tun hatten und die ihnen bereits von den vorangegangenen Problemstellungen bekannt sind [Zan07]. Diese Vorgehensweise behindert allerdings das Finden neuer und alternativer Konzepte.

Für die Auswahl von Antriebssystemen bzw. den Einsatz von Direktantrieben, gibt

es im *Wachstums-kern VERDIAN*<sup>1</sup> [Ver09] bisher nur einen Prototypen als Assistenz-Werkzeug. Dieser Prototyp ist in Form eines *Funktion-Struktur-Speicher (FSS)* (siehe Kapitel 2.2) in einer Datenbank umgesetzt. Des Weiteren gibt es ein rudimentäres Auswahltool, welches die Suche und Auswahl nach qualitativen Eigenschaften von Aktoren ermöglicht. Bisher wurden für die Auswahl von Aktoren immer klassische Konstruktions- und Produktkataloge herangezogen (vgl. [Erb09]).

## 1.2 Ziel der Arbeit

Ziel dieser Arbeit ist die Entwicklung und Implementation einer Webanwendung für die Suche von Direktantrieben sowie deren Ergebnisdarstellung. Bei der Entwicklung liegt der Schwerpunkt auf der Gestaltung des UI der Webanwendung, den Interaktionsmöglichkeiten mit dem Nutzer und den geeigneten Visualisierungen zur Ergebnisdarstellung. Dabei steht die *Usability* (siehe Kapitel 2.3) der Webanwendung im Vordergrund. Die Gestaltung basiert auf generellen Designprinzipien (Kapitel 3.5), welche an die Anforderungen potentieller Nutzer mit Hilfe einer Anforderungsanalyse (siehe Kapitel 3) angepasst wurden.

Aufgrund der sich häufig widersprechenden Eigenschaften von Aktoren, kann es bei der Suche nach dem geeigneten Aktor zu folgender Problematik kommen. Auf der einen Seite ist es möglich, dass die Ergebnismenge zu sehr eingeschränkt werden kann und somit keine (in der Literatur als *Zerohit-Problem* [Rei05] bekannt) oder nur wenige Aktoren als Ergebnisse übrig bleiben; auf der anderen Seite kann es sein, dass es zu viele Aktoren als Ergebnisse gibt (in der Literatur als *Megahit-Problem* bekannt), bei denen die Unterschiede nicht deutlich werden und folglich keine konsistente Auswahl getroffen werden kann. Ziel dieser Arbeit ist es demzufolge, eine Webanwendung zu erstellen, die dieses Dilemma durch geeignete Suchmechanismen, Visualisierung für die Ergebnisse und einfachen Interaktionsmöglichkeiten löst.

Im *Wachstums-kern VERDIAN* [Ver09] existiert bereits eine Datenbank mit Aktoren

---

<sup>1</sup>Vernetzung von Unternehmen und Forschungseinrichtungen der Rennsteigregion sowie die *TU Ilmenau* und die *IMMS gGmbH*, die auf dem Gebiet der Direktantriebstechnik zusammenarbeiten

und ihnen zugeordneten Eigenschaften, deren Bestand immer weiter vergrößert wird. Diese Datenbank bildet die Grundlage für den genutzten Suchalgorithmus und das dazu aufgesetzte, zu entwickelnde Frontend. Eine parallel laufende Arbeit zu diesem Thema beschäftigt sich ausschließlich mit dem Suchalgorithmus und dessen Abstandsmaß für unscharfe Auswahl von Direktantrieben.

## 1.3 Aufbau der Arbeit

Der Aufbau und die Vorgehensweise dieser Arbeit orientieren sich am *Usability Engineering Livecycle (UEL)* nach Mayhew [May99]. Nachdem zu Anfang eine Einleitung erfolgte, werden danach die Grundlagen erläutert. Das anschließende Kapitel, welches die erste Phase des UEL bildet, beinhaltet die Anforderungsanalyse und fasst die grundlegenden Anforderungen an die zu entwickelnde Webanwendung zusammen. Als Nächstes werden Visualisierungs- und *Graphical User Interface (GUI)*-Ideen diskutiert sowie bewertet und wichtige Designentscheidungen getroffen. Im Anschluß wird die Entwicklung, d. h. Implementation, der Webanwendung beschrieben. Dabei werden verwendete Web-Technologien kurz erklärt, verfügbare Visualisierungs-Frameworks und Bibliotheken verglichen und deren Umsetzung erläutert. Abschließend wird eine Zusammenfassung und ein Ausblick zu dieser Arbeit gegeben.

# Kapitel 2

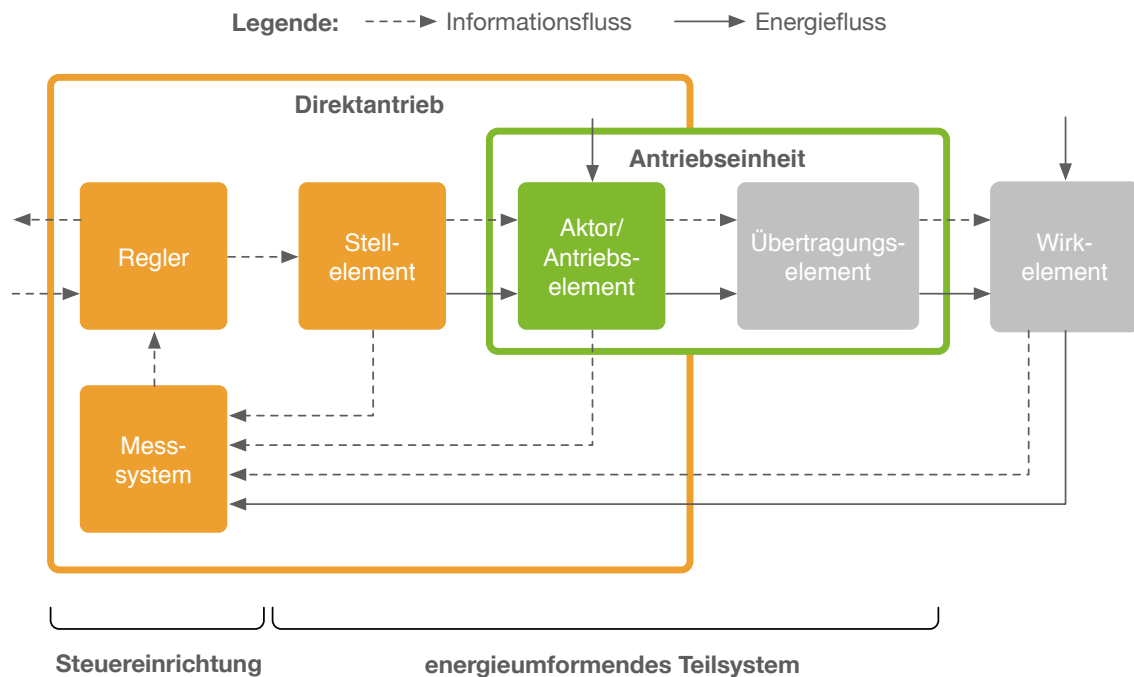
## Grundlagen

In diesem Kapitel sollen wichtige Termini aus dem Bereich mechatronischer Systeme und der Usability erklärt werden. Diese Begriffe sollen dabei helfen, ein besseres Verständnis über wesentliche Aspekte und Entscheidungen in dieser Arbeit zu erhalten.

### 2.1 Begriffsabgrenzung Aktor und Direktantrieb

Das Antriebselement oder auch **Aktor** genannt, bezeichnet ausschließlich den elektromechanischen Energiewandler. Dieser formt die elektrische (nicht mechanische) Energie aus dem Stellelement in mechanische Arbeit (Bewegungsenergie) um. Der Aktor befindet sich in der Wirkungskette eines mechatronischen Systems zwischen der Regelungs- bzw. Steuereinrichtung und dem Wirkelement [Zan07] (siehe Abbildung 2.1). Die Wirkung eines Antriebsvorganges drückt sich in der gewünschten Bewegung des Wirkelementes aus. Sie stellen in jedem mechatronischen System eine unverzichtbare Komponente zur Realisierung von Bewegungsaufgaben dar.

Der **Direktantrieb** dagegen bezeichnet ein Antriebssystem ohne den mechanischen Energiewandler, d. h. der Aktor wirkt direkt, ohne Getriebe (Übertragunselement), auf ein Wirkelement. Heutzutage werden zunehmend Direktantriebe verwendet, da sie gegenüber konventionellen Antrieben Vorteile bezüglich der Genauigkeit, Dynamik und der kompakten Bauform aufweisen (vgl. [Win08], [Ver09]).



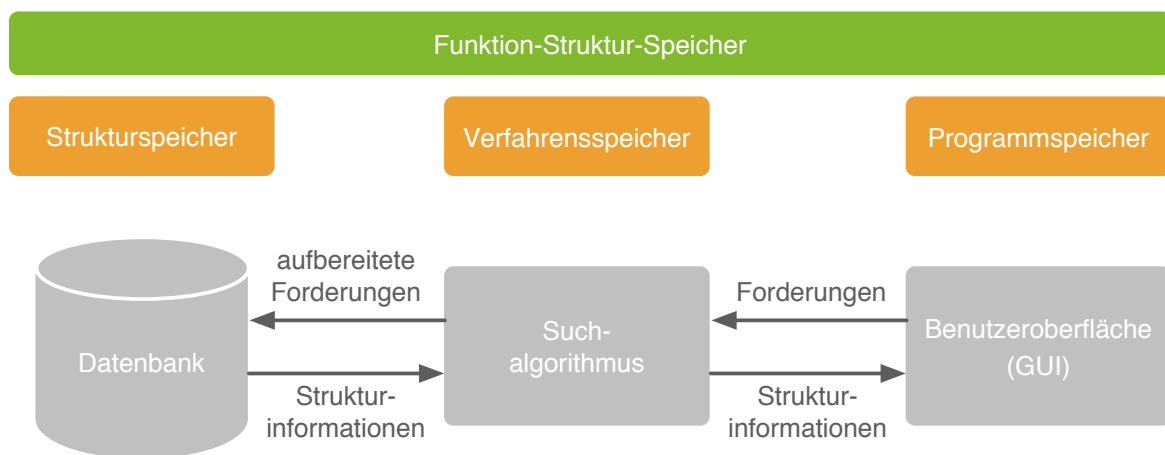
**Abbildung 2.1:** Struktur eines Antriebssystems (vgl. [KB91], [Zan07])

## 2.2 Funktion-Struktur-Speicher

Der Funktion-Struktur-Speicher (FSS) ist ein Speichersystem, welches nach Angabe von gewünschten Funktionen und Eigenschaften eine oder mehrere Strukturen vorgibt. Der Anwender kann somit durch die Vorgabe von funktionellen bzw. strukturellen Eigenschaften Lösungsansätze heraussuchen, die für seine spezifische Antriebsaufgabe in Frage kommen.

Der FSS gliedert sich im *Konstruktiven Entwicklungsprozess (KEP)* nach der Zerlegung der Gesamtfunktion in die Teilfunktionen ein (vgl. [Han76], [Zan07]). Er enthält Strukturdaten von Aktoren, welche dazu dienen, dem Entwickler die Objekte anzuzeigen, welche für die Lösung der Bewegungsaufgabe in Betracht kommen. Dabei können unterschiedliche Aktorprinzipien vorgeschlagen werden. Es besteht auch die Möglichkeit, dass dieses Speichersystem Strukturen vorgibt, die der Anwender bei seiner vorhergehenden Antriebsauslegung nicht in Betracht gezogen hat. Dadurch ist es möglich, dass neue Antriebskonzepte generiert werden. Der FSS kann auch zur Evaluierung bereits

ausgewählter Antriebskonzepte dienen. Dabei soll die Lösungsmenge der potentiell geeigneten Antriebsprinzipien schon in der Konzeptphase auf eine überschaubare Anzahl eingeschränkt werden. Ziel dieses FSS ist es, eine übersichtliche Menge an Lösungsmöglichkeiten bereitzustellen, mit denen die gestellte Aufgabe realisierbar ist, statt eine ideale Lösung für die Antriebsaufgabe zu finden.



**Abbildung 2.2:** Grundkonzept des Funktion-Struktur-Speichers (vgl. [Han76], [Win08])

## 2.3 Usability

### Definition

Usability oder auch Gebrauchstauglichkeit ist nach der ISO 9241-11 als

“das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufrieden stellend zu erreichen”

festgelegt [DIN03].

Usability wird also durch die drei Hauptkriterien: *Effektivität*, *Effizienz* und *Zufriedenheit* dargestellt.

- *Effektivität* beschreibt, ob der Benutzer einer Software oder einer Webseite seine Aufgaben überhaupt ausführen kann, um somit seine Ziele zu erreichen.
- Das Kriterium *Effizienz* bezieht sich auf den Aufwand, der zum Erreichen des Ziels nötig ist respektive, ob sich die Aufgaben mit angemessenem bzw. möglichst geringem Aufwand erledigen lassen.
- *Zufriedenheit*, welche die subjektive Komponente bildet [HV03], gibt an, ob ein Nutzer die Software mag und sich damit wohl fühlt. Benutzer müssen demzufolge auch subjektiv den Eindruck haben, die Software effektiv und effizient zu bedienen.

*Jakob Nielsen* [Nie93] beschreibt Usability in einer ähnlichen Form. Von ihm wird Usability durch folgende fünf Komponenten beschrieben:

- *Erlernbarkeit*: Die Software sollte leicht erlernbar sein.
- *Effizienz*: Die Software sollte effizient zu nutzen sein, damit erfahrene Nutzer möglichst produktiv mit ihr arbeiten können.
- *Einprägsamkeit*: Die Software sollte leicht einprägsam sein, damit der Nutzer sich nicht immer wieder neu einarbeiten muss, wenn er die Software mal einige Zeit nicht genutzt hat.
- *Fehler*: Außerdem sollte die Software eine niedrige Fehlerrate haben, damit beim Nutzen möglichst wenig Fehler auftreten.
- *Zufriedenheit*: Die Nutzung der Software sollte angenehm zu bedienen sein, damit Nutzer Spaß haben beim Bedienen.

Es gibt viele weitere Definitionen für Usability (siehe [HV03], [RC01]), auf die hier nicht näher eingegangen wird.

Abschließend soll hier noch eine Charakterisierung nach [May99] geliefert werden. *Mayhew* beschreibt Usability als messbare Charakteristik des *User Interface (UI)* eines Produktes und meint damit wie leicht das UI zu erlernen und zu nutzen ist.

## Notwendigkeit

Usability wird oft nur als Modewort benutzt. Die Wirkung, welche es auf ein Produkt, eine Software oder ein interaktives System hat, wird oftmals unterschätzt. Usability Tests sollten nicht erst kurz vor Auslieferung bzw. Fertigstellung eines Produktes durchgeführt werden. Die Usability-Probleme, welche dann aufgedeckt werden, sind in dem kurzen Zeitraum bis zur Veröffentlichung gar nicht mehr oder nur mit sehr hohem finanziellen Aufwand lösbar. Usability muss in den Entwicklungsprozess eingegliedert werden.

Ein Usability-orientierter Entwicklungsprozess erhöht die Wahrscheinlichkeit der späteren Nutzerakzeptanz und verringert zudem Kosten (Supportkosten, Kosten für Dokumentation und Training, etc.). Die Anforderungen des Nutzers stehen während des ganzen Entwicklungsprozesses im Vordergrund. Die Vorteile des Nutzers sind: höhere Produktivität, kürzerer Anlernzeit und weniger Fehler des Nutzers bei der Bedienung, was zu weniger Frustration und mehr Spaß führen kann. Die Vorteile auf Seiten des Entwicklers bzw. Herstellers sind: höhere Gewinne, wettbewerbsfähigere Produkte und Services, verringerte Entwicklungs- und Wartungskosten und geringere Kundensupportkosten, um nur einige zu nennen [May99]. Eine ausführlichere Beschreibung sowie weitere Vorteile, welche Usability mit sich bringt, sind in [HV03] dargestellt.

Da die Leistungsfähigkeit eines Systems nicht nur von der verwendeten Technik, sondern vor allem von der effizienten Interaktion zwischen Mensch und Maschine abhängig ist, müssen die individuellen Anforderungen der einzelnen Nutzer bei der Entwicklung mit einbezogen werden. Eine Studie der *Standish Group* hat gezeigt, dass unvollständige Anforderungen und mangelnder Einbezug der Nutzer die Hauptgründe sind, weshalb Software-Projekte scheitern (siehe [RF07, S. 9]).



## 2.4 Usability Engineering Lifecycle

Für die Entwicklung von interaktiven Systemen (Software, Webseiten, -anwendungen etc.) bietet sich das Modell des Usability Engineering Lifecycle (UEL) von *Mayhew* [May99] an. Dieser Usability-orientierte Entwicklungsprozess ist speziell auf Prozesse der Softwareentwicklung zugeschnitten und bindet Usability-Methoden in den Entwicklungsprozess mit ein. Ziel ist das Erreichen einer besseren Usability im zu entwickelnden Produkt. Eine gutes UI-Design bildet die Grundlage für eine effiziente und effektive Benutzung des Produktes. Der UEL ist ein iterativer Prozess und beschreibt eine Folge logisch zusammenhängender Aktivitäten zum Erreichen einer besseren Usability. Die dort verwendeten Methoden stammen aus mehreren anderen Fachgebieten, wie z.B. kognitive Psychologie, experimentelle Psychologie, Ethnographie und Software Engineering [Bra06]. Der UEL stellt eine Richtlinie für den Entwicklungsprozess dar und sollte nicht als unflexible Abfolge von Regeln verstanden werden. Es gibt eine Vielzahl von Techniken, die zur Ausführung der einzelnen Arbeitsschritte genutzt werden können. Welche davon wie eingesetzt werden, liegt letztendlich in der Entscheidung des Usability-Experten.

Der UEL besteht aus drei Phasen.

1. Anforderungsanalyse
2. Design/Test/Entwicklung
3. Installation

Abbildung 2.3 verdeutlicht, wie diese drei Phasen und ihre zugehörigen Aktivitäten zusammenhängen. Nachfolgend sollen diese drei Phasen kurz näher beschrieben werden.

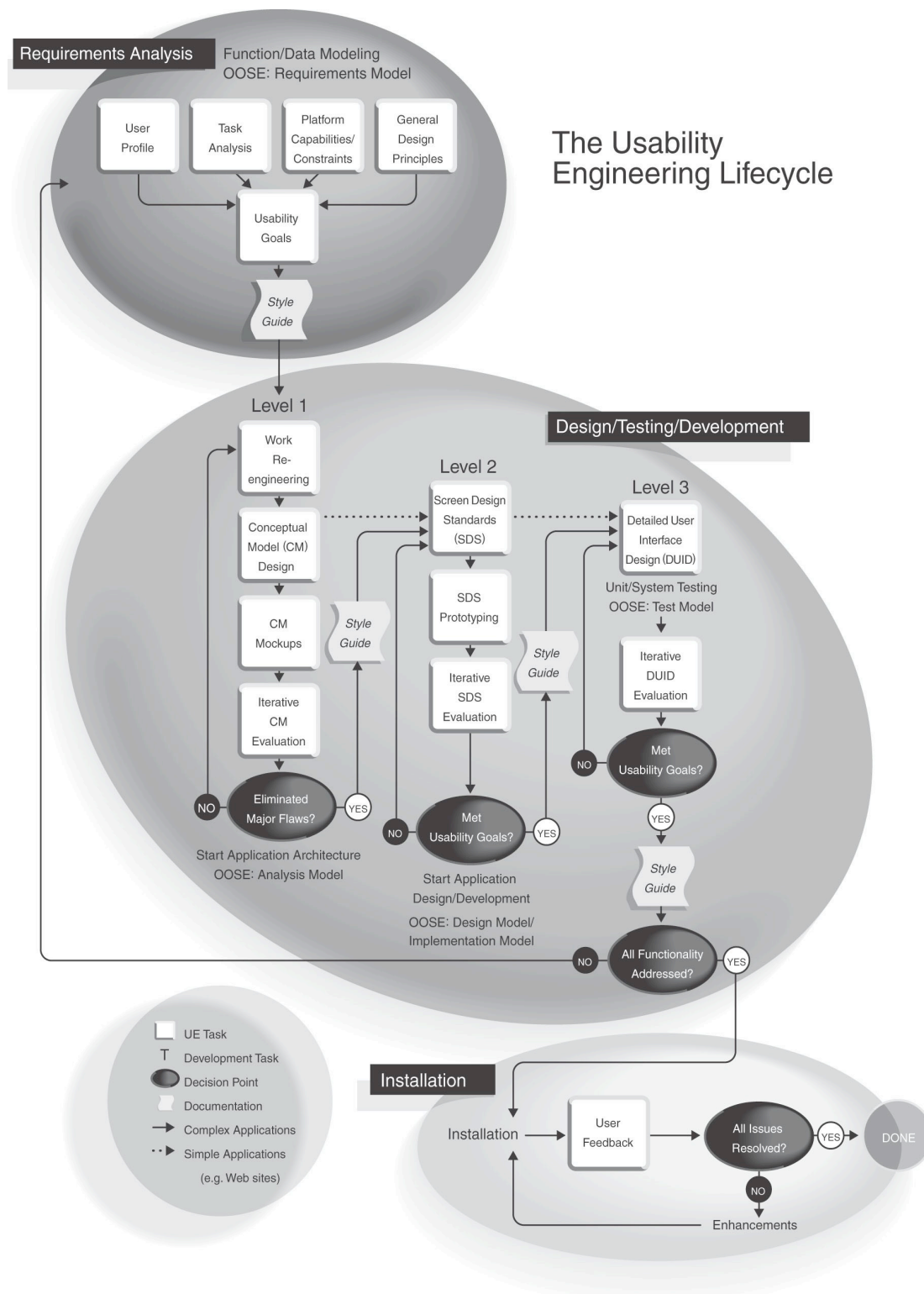


Abbildung 2.3: Überblick Usability Engineering Lifecycle [May99]

## 1. Phase: Anforderungsanalyse

Die Anforderungsanalyse hilft dem Entwickler, eine Vorstellung von den Nutzern und deren Aufgaben zu bekommen. Außerdem werden in dieser Phase Einschränkungen und Anforderungen an das System spezifiziert. Ziel der Anforderungsanalyse ist es, Usability-Ziele zu definieren, mit deren Hilfe die Evaluation in späteren Phasen stattfinden kann. Die Anforderungsanalyse gliedert sich in die Teilphasen: Nutzerprofilierung, Aufgabenanalyse, Plattformbeschränkungen, Designprinzipien und Usability Ziele, die im folgenden näher erläutert werden.

**Nutzerprofil:** In diesem Schritt werden spezifische Nutzermerkmale für die anvisierte Nutzergruppe aufgestellt, welche später maßgeblich das UI-Design beeinflussen. Z. B. haben Studenten meist eine größere Affinität bzgl. Computern und Software, als Professoren. Das heißt, dass Studenten sich schneller an neue Bedienkonzepte und UI gewöhnen und Professoren dafür evtl. eine längere Einarbeitungszeit benötigen.

**Aufgabenanalyse:** Sie dient zur Erfassung der Ziele, Aufgaben und Arbeitsabläufe potentieller Nutzer. Es geht darum, das *mentale Modell*<sup>1</sup> der Nutzer zu verstehen, um daraus Eigenschaften für das UI-Design abzuleiten. Bezogen auf die Suche nach Akteuren ist es wichtig zu erfahren, wie Ingenieure bisher vorgegangen sind, um geeignete Akteure zu finden. Dem Ingenieur bereits bekannte Modelle und Metaphern können dann evtl. beim UI-Design berücksichtigt werden.

**Plattformbeschränkungen:** In dieser Aktivität wird das Potential und die Einschränkungen, welche die technologische Plattform (Betriebssystem, Browser, Internet) für das UI-Design bieten, bestimmt und dokumentiert. Z. B. muss beachtet werden, dass eine Webanwendung größeren Einschränkungen unterliegt als ein Programm für ein bestimmtes Betriebssystem.

---

<sup>1</sup>Spiegelt das Verständnis der Funktionsweise eines Systems wieder

**Designprinzipien:** In dieser Phase werden relevante Richtlinien und Prinzipien zum UI-Design gesammelt und überprüft und während des Designprozesses angewandt.

**Usability-Ziele:** In diesem Schritt werden spezifische qualitative Ziele bestimmt. Diese spiegeln die Anforderungen aus dem Nutzerprofil und der Aufgabenanalyse wider. Weiterhin werden quantitative Ziele definiert, welche als Kriterien in Usability-Tests genutzt werden. Die festgelegten Usability-Ziele dienen der Evaluation in der späteren Phase des UEL.

Die Ergebnisse der Anforderungsanalyse können in einem so genannten Product Style Guide dokumentiert und somit später bei weiteren Versionen der Software oder neuen ähnlichen Produkten immer wieder verwendet werden.

## 2. Phase: Design/Test/Entwicklung

Die zweite Phase des UEL ist in drei Ebenen eingeteilt und zeichnet sich durch einen Top-Down Ansatz im UI-Design aus. Die drei Ebenen lassen sich wie folgt beschreiben:

- In der ersten Ebene, dem *Conceptual Design*, wird das konzeptionelle High-Level Design festgelegt und es finden erste Tests mit Papierprototypen (Mock-Ups) statt.
- Danach erfolgt der Entwurf von Richtlinien und Designstandards für das produktspezifische *Screen-Design*. Des Weiteren wird der aus dem vorherigen Schritt erstellte Prototyp zu einem Screen-Design weiterentwickelt; d. h., es wird ein detaillierter Prototyp für die Oberfläche entworfen. Dieser soll die festgelegten Designstandards verdeutlichen.
- In der dritten Ebene wird das komplette, *detaillierte UI-Design* fertiggestellt. Grundlage dafür bilden die vorangegangenen Schritte.

Erst nach dem konzeptionelle Entscheidungen getroffen und evaluiert wurden, wird sich den Details des UI-Designs zugewandt. Alle drei Ebenen bestehen aus einem iterativen

Prozess von Tests und Redesigns. Dieses Vorgehen gewährleistet die Einbeziehung des Nutzers bei der Konzeption aller Ebenen.

### **3. Phase: Installation**

Nachdem das Produkt implementiert und einige Zeit genutzt wurde, kann von dem Nutzer Feedback geholt werden, um das UI-Design noch weiter zu verbessern und um evtl. neue Versionen oder ähnliche Produkte darauf aufzubauen. Da diese Phase eine gewisse Zeit der Inbetriebnahme der zu erstellenden Webanwendung voraussetzt, wird dieser Schritt nicht berücksichtigt. Er kann aber dennoch im Rahmen einer weiteren Arbeit zu einem späteren Zeitpunkt nachgeholt werden.

## Kapitel 3

# Anforderungsanalyse

Die Anforderungsanalyse, in der alle relevanten Rahmenbedingungen für die zu erstellende Webanwendung für die Aktorsuche festgelegt werden, bildet den ersten Schritt im UEL. In dieser Phase werden die potentiellen Nutzergruppen und ihre Anforderungen definiert, ihre Arbeitsaufgaben analysiert, Plattformbeschränkungen und Designprinzipien konsolidiert. Abbildung 2.3 zeigt einen Überblick der einzelnen Aktivitäten dieser Phase.

### 3.1 Nutzerprofil

Es gibt kein “Bestes” UI-Design, welches jedem Nutzer gerecht wird [May99, S. 35]. Jeder Nutzer bringt andere Voraussetzungen mit und hat deshalb andere Anforderungen an die Bedienung der Webanwendung. Manche UI-Designs tragen zu mehr Effektivität bei der Nutzung für den einen Nutzertyp bei, für einen anderen wirkt es sich allerdings eher ineffektiv aus. Während Nutzer, welche die Webanwendung zum ersten Mal oder nur sehr unregelmäßig nutzen, sich ein einfaches UI wünschen, möchten Profi Nutzer wahrscheinlich ein UI, welches Ihnen viele Einstellmöglichkeiten bietet. Damit ein UI erstellt werden kann, welches für möglichst viele Nutzer bzw. Nutzertypen gut einsetzbar ist, müssen vorab die unterschiedlichen Nutzertypen gefunden und analysiert werden.

Es gibt zwei Möglichkeiten Nutzerprofile zu beziehen:

1. **Fragebogen**, der an Nutzer ausgeteilt wird.
2. **Experteninterviews**, d. h. Interviews mit Personen, die über zukünftige Nutzer Bescheid wissen.

Auch wenn die erste Methode die zuverlässigeren und akkurateren Ergebnisse ermöglicht, ist sie für Webseiten generell nicht praktikabel. Nach *Mayhew* ist es zulässig, Aussagen zu den Nutzerprofilen aufgrund von Experteninterviews abzuleiten [May99, S. 45]. Es wurden somit Experteninterviews mit den Auftraggebern bzw. Projektbetreuern durchgeführt und die Nutzerprofile erstellt.

Folgende Nutzerprofile konnten letztendlich identifiziert werden:

- Studenten
- Ingenieure aus Universität
- Ingenieure aus der freien Wirtschaft
- Projektleitung

Tabelle 3.1 zeigt diese Nutzertypen und ihre grundlegenden Charakteristika. Wie daraus zu erkennen ist, handelt es sich bei den Nutzern um recht homogene Nutzertypen.

## 3.2 Aufgabenanalyse

### 3.2.1 Zielsetzung und Vorgehensweise

Nach [May99] geht es bei der Aufgabenanalyse darum, die aktuellen Arbeitsweisen der Nutzer zu analysieren und auf das zu entwickelnde System zu übertragen. Während dieser Phase sollen Nutzerziele spezifiziert und ein nutzerzentriertes Modell der Arbeitsaufgaben und -abläufe erstellt werden.

Tabelle 3.1: Überblick Nutzerprofile

	Studenten	Ingenieure aus Universität	Ingenieure aus freien Wirtschaft	Projektleitung
<b>Physische Eigenschaften</b>				
Alter	19-25	25-66	25-66	25-66
Geschlecht	hauptsächlich männlich	hauptsächlich männlich	hauptsächlich männlich	hauptsächlich männlich
Ausbildung	Abitur	abgeschlossenes Ing.-Studium	abgeschlossenes Ing.-Studium	technisches/naturwissenschaftliches Studium
<b>Einstellung / Motivation</b>				
Aufgeschlossenheit gegenüber Computern	sehr gut	sehr gut	gut bis sehr gut	gut
Aufgeschlossenheit gegenüber Internetanwendungen	sehr gut	sehr gut	gut bis sehr gut	gut
Bereitschaft zum Einarbeiten in neue Software	hoch bis sehr hoch	hoch	hoch	hoch
<b>Kenntnisse / Erfahrungen</b>				
Computer Kenntnisse	sehr gut	sehr gut	gut bis sehr gut	gut bis sehr gut
Häufigkeit der Nutzung von Computern	sehr häufig	häufig	häufig	häufig
Grund für Nutzung von Computern (nach Häufigkeit)	<ul style="list-style-type: none"> <li>• Kommunikation</li> <li>• Recherche</li> <li>• Textverarbeitung</li> </ul>	<ul style="list-style-type: none"> <li>• Kommunikation</li> <li>• Recherche</li> <li>• Textverarbeitung</li> <li>• Simulation</li> </ul>	<ul style="list-style-type: none"> <li>• Kommunikation</li> <li>• Recherche</li> <li>• Textverarbeitung</li> <li>• Simulation</li> </ul>	<ul style="list-style-type: none"> <li>• Kommunikation</li> <li>• Koordination</li> <li>• Planung</li> </ul>
Internet Kenntnisse	sehr gut	gut bis sehr gut	gut	gut
Häufigkeit der Nutzung des Internets	sehr häufig	häufig	häufig	häufig
<b>Ausstattung / Umgebung</b>				
Arbeitsumgebung	Büroarbeitsplatz, gute Konzentrationsmöglichkeiten	Büroarbeitsplatz, gute Konzentrationsmöglichkeiten	Büroarbeitsplatz, gute Konzentrationsmöglichkeiten	Büroarbeitsplatz, gute Konzentrationsmöglichkeiten
Hardware Ausstattung	<ul style="list-style-type: none"> <li>• sehr schnelle Internetverbindung</li> <li>• sehr gute Rechenleistung</li> <li>• ab 17" Monitore ab 1440x900px</li> </ul>	<ul style="list-style-type: none"> <li>• sehr schnelle Internetverbindung</li> <li>• sehr gute Rechenleistung</li> <li>• ab 17" Monitore ab 1440x900px</li> </ul>	<ul style="list-style-type: none"> <li>• sehr schnelle Internetverbindung</li> <li>• sehr gute Rechenleistung</li> <li>• 19"-24" Monitore ab 1440x900px</li> </ul>	<ul style="list-style-type: none"> <li>• sehr schnelle Internetverbindung</li> <li>• sehr gute Rechenleistung</li> <li>• Notebookdisplays ab 1024x768px</li> </ul>



Die grundlegende Vorgehensweise bei der Aufgabenanalyse ist (nach [May99]):

- Erfassen von Hintergrundinformationen zu den Aufgaben,
- Ideensammlung,
- Strukturierte Interviews mit Nutzern,
- Beobachten von Nutzern im Arbeitsalltag,
- Erstellen eines Modells zu den Arbeitsabläufen des Nutzers.

Bei der Entwicklung von Webseiten bzw. einer Webanwendung ist diese Vorgehensweise allerdings nur sehr eingeschränkt möglich. Deshalb liegt dort der Fokus mehr auf dem Zusammenstellen der Wünsche und Bedürfnisse der Nutzer bzgl. zukünftiger Funktionalität [May99, S. 104f]; d. h., welche Aufgaben möchte der Nutzer mit Hilfe der Anwendung erledigen; wie stellt er sich den Umgang mit der Anwendung vor. Diese Wünsche und Bedürfnisse werden durch das mentale Modell des Nutzers widerspiegelt.

Außerdem gilt es herauszufinden, wie die Nutzer aktuell ihre Aufgaben ausüben; d. h., wie sie bei der Suche nach Aktoren vorgehen. Diese bestehenden Vorgehensweisen müssen vom UI der Anwendung abgebildet werden. Dadurch kann später Eingewöhnungszeit und Trainingsaufwand bei der Benutzung der neuen Anwendung minimiert werden. Weiterhin gilt es Arbeitsabläufe effizient zu gestalten. Je näher das UI dem mentalen Modell der zukünftigen Nutzer der Webanwendung zur Suche nach Aktoren kommt, desto höher ist auch die Usability dieser Webanwendung.

Mit Hilfe von Experten- und Nutzer-Interviews wurden Hintergrundinformationen zu den bisherigen Arbeitsabläufen bei der Suche nach Aktoren und dem mentalen Modell der Nutzer erfasst. Die Experten-Interviews wurden mit den Betreuern des Projektes geführt. Diese dienten als repräsentative Nutzer, da sie am besten über die zukünftigen Nutzergruppen und ihre Aufgaben Bescheid wissen. Ein weiteres Interview wurde mit zwei Personen von *Carl Zeiss Meditec Vertriebsgesellschaft mbH, Jena* geführt. Diese waren repräsentativ für die Nutzergruppe "Ingenieure aus der freien Wirtschaft". Des

Weiteren wurde ein Fragebogen an Studenten und Ingenieure an der *TU Ilmenau* verteilt, welcher zehn Rückmeldungen ergab.

### 3.2.2 Ergebnisse

Der folgende Teil enthält die Zusammenfassung aus den geführten Interviews und den Fragebögen mit den zukünftigen Nutzern. Daraus geht hervor, dass keine exorbitanten Wünsche oder Anforderungen an die Suche und Ergebnisdarstellung vorliegen. Lediglich der Stand der Technik bzw. die aus dem Internet bekannten Mechanismen zum Suchen und Auswählen sollten vorhanden sein. Im Folgenden soll nun die bisherige Vorgehensweise, bei der Suche nach Aktoren und das mentale Modell der Nutzer beschrieben werden.

#### Bisherige Vorgehensweise

Die bisherige Vorgehensweise bei der Suche nach Aktoren bzw. Direktantrieben ist bei den Ingenieuren größtenteils gleich. Für die Suche werden das Internet, Bücher (Fachliteratur), Hersteller-Webseiten und -Kataloge, Datenblätter und der eigene Erfahrungsschatz genutzt. Außerdem wird sich unter den Kollegen bzgl. Erfahrungen und Meinungen ausgetauscht oder direkt Anfragen per Telefon oder Mail an den jeweiligen Hersteller gerichtet. In der Regel sieht das Vorgehen wie folgt aus:

1. Vorauswahl des erforderlichen Antriebsprinzips mittels Fachliteratur,
2. Evtl. Suche in vorrätigen Hersteller-Katalogen (falls vorhanden),
3. Evtl. Suche auf gängigen Hersteller-Webseiten (falls bekannt),
4. Google Suche zu möglichen Anbietern (wer liefert was),
5. Suche auf gefundenen Hersteller-Webseiten,
6. Einsichtnahme in Datenblätter der Hersteller,
7. Auswahl eines Aktors.

Weiterhin werden auch Produkt- und Lieferanten-Suchmaschinen wie [www.wlw.de](http://www.wlw.de) und [www.directindustry.de](http://www.directindustry.de) verwendet. Aus den Interviews und dem Fragenkatalog ging weiterhin hervor, dass eine Suche nach Antrieben in diskontinuierlichen Abständen in der Regel ein bis sechs Mal im Jahr gemacht wird. Die Suche nach Aktoren war bisher immer sehr zeitintensiv. Eine solche Suche konnte durchaus drei bis vier Stunden und unter Umständen sogar noch länger dauern. Von zehn befragten Ingenieuren haben sechs schon einmal eine Expertensuche genutzt und kennen sich somit mit der Suche nach erweiterten Suchkriterien aus. Vier Ingenieure nutzten bisher immer nur die einfache Suche oder wussten mit Expertensuche nichts anzufangen. Als verwendeter Browser wurde am häufigsten der *Mozilla Firefox* angegeben (7x). An zweiter Stelle kam der *Microsoft Internet Explorer (IE) 7* (5x) und danach der *Opera* Browser (3x). Das Schlusslicht bildete der *IE 6* (2x). Dabei war es auch möglich, Mehrfachnennung anzugeben. Andere Browser, wie z. B. *Safari* oder *Google Chrome*, wurden nicht angegeben. Insgesamt bedeutet das, bei der Entwicklung der Webanwendung muss besonders beim *Firefox*, *IE 7* und *Opera* eine korrekte Darstellung gewährleistet werden. Browserweichen für den *IE 6* müssen nicht unbedingt implementiert werden. Bezüglich der gewünschten Zeitvorgabe für die zukünftige Suche nannten die Befragten, die Suche in möglichst minimaler Zeit durchführen zu wollen. Prinzipiell sollte das aber nicht länger dauern als bei ihrer bisherigen Vorgehensweise.

### Suchanfrage

Die Analyse und Auswertung der Interviews und Fragebögen ergab, wie sich die zukünftigen Nutzer das Suchinterface für die Aktorsuche vorstellen. Das UI für die Suche sollte eine schrittweise Einschränkung möglich machen. Dabei muss es möglich sein, alle verfügbaren Parameter zur Einschränkung der Suche auswählen zu können. Allerdings muss die Übersichtlichkeit trotzdem gewährleistet sein. Die Übersichtlichkeit des Suchinterfaces wird von allen Nutzern als sehr wichtig empfunden. Manche Nutzer wünschen sich außerdem, dass die Parameter nach -Familien zusammengefasst sind. Teilweise war auch eine Gewichtung der Parameter gewünscht. Außerdem sollten nicht alle Parameter bei der Eingabe zwingend verlangt werden, um eine Suche auszuführen.

Während man die einzelnen Kriterien für die Suche wählt und somit die Suche einschränkt, muss angezeigt werden, wie viele Ergebnisse übrig bleiben. Weiterhin wird verlangt, die Suchanfrage mit der Enter-Taste auszulösen. Im Allgemeinen sollte die Bedienung mit der Tastatur unterstützt werden, um auch Menschen mit körperlichen Behinderungen eine Chance zu geben, die Webanwendung zu bedienen (Barrierefreies Internet<sup>1</sup> siehe dazu [Hel06]).

### **Ergebnisdarstellung**

Bei der Ergebnisdarstellung wünschen sich die Nutzer eine einheitliche, einfache und übersichtliche Darstellung. Diese sollte das Vergleichen der verschiedenen Aktoren leicht möglich machen. Bisher ist ein Vergleich untereinander immer sehr zeitintensiv, besonders bei Aktoren verschiedener Hersteller. Eine tabellarische Darstellung der Ergebnisse würde diese Funktionalität begünstigen. Außerdem wurde die Möglichkeit zum Sortieren und weiteren Filtern der Ergebnisse gewünscht. Bildhafte Darstellungen könnten für das schnellere Erfassen der Ergebnisse genutzt werden. Ein weiterer Wunsch ist die Verlinkung der Hersteller-Webseite des jeweiligen Aktors. Dadurch könnte man an weitere Daten oder Bestellinformationen gelangen. Die Ergebnisdarstellung sollte es auch möglich machen, wichtige Parameter farblich hervorzuheben und die Grenzwerte der Kriterien anzuzeigen. Weitere technische Informationen (z. B. Zeichnungen, Kennlinien etc.) sollten auf Wunsch abrufbar sein.

### **Feedback während der Suche**

Wenn die Suche keine oder sehr viele Ergebnisse liefert, ist es wichtig, dass die Webanwendung dem Nutzer rechtzeitig geeignetes Feedback liefert. Dies kann dadurch geschehen, dass nach jeder Angabe eines Kriteriums, die Anzahl der verbleibenden Ergebnisse gut sichtbar im Suchinterface angezeigt wird. Liefert die Suche nicht die gewünschte Anzahl der Ergebnisse, muss es leicht möglich sein, sie mehr oder we-

---

<sup>1</sup>bezeichnet Webseiten, die von allen Nutzern unabhängig von körperlichen oder technischen Möglichkeiten, uneingeschränkt, d. h. barrierefrei genutzt werden können

niger einzuschränken. Dabei sollten die voreingestellten Parameter gemerkt werden und leicht änderbar sein. Ein weiterer Wunsch für geeignetes Feedback besteht darin, dass die Webanwendung automatisch Kriterien für die Einschränkung der Suche vorschlägt. Wenn es keine Ergebnisse gibt, die alle Kriterien erfüllen, sollten wenigstens die Ergebnisse, die in der Nähe liegen, aufgelistet bzw. dargestellt werden. Durch die Implementation des *Skyline-Algorithmus* (siehe [BKS01]) wird dies mit Hilfe von Relevanzen gelöst. Wenn es zu viele Ergebnisse gibt, sollte die Webanwendung die relevantesten Ergebnisse zuerst anzeigen und das Sortieren nach anderen Parametern erlauben.

### Sonstige Funktionalitäten

In den Fragebögen wurden die Ingenieure außerdem interviewt, welche weiteren Funktionalitäten sie sich von der Webanwendung zur Suche nach Aktoren wünschen. Es wurde sich ein Nutzer-abhängiges Bewertungssystem für die Aktoren gewünscht sowie die Möglichkeit, Suchanfragen abzuspeichern, um zu einem späteren Zeitpunkt weiter zu suchen. Auch das Ausdrucken der Suchanfrage und seiner Ergebnisse sollte möglich sein. Weiterhin sollten die Suche und Vergleiche der Aktoren herstellerübergreifend erfolgen. Bei den Ergebnissen sollten Links zu den Kontaktdaten des Herstellers sowie zu Datenblättern aufgeführt werden.

### Eigenschaften der Suche

Weiterhin wurden die Nutzer gefragt, mit welchen Eigenschaften sie ihre Vorstellungen und Ansprüche an die Webanwendung beschreiben würden.

Dabei haben sich folgende Eigenschaften ergeben (alphabetische Reihenfolge):

- *Eindeutig*: Die Suchergebnisse sollten möglichst eindeutig und treffsicher sein.
- *Erweiterbar*: Die Funktionalitäten der Aktoren-Suche sollten erweiterbar sein.
- *Hierarchisch*: Die Eingrenzung der optimalen Lösung sollte hierarchisch (trichterförmig) geschehen.
- *Interaktiv*: Die Suche und Visualisierung soll interaktiv bedienbar sein.

- *Intuitiv*: Die Bedienung der Webanwendung muss intuitiv erfolgen können, ohne dass eine extra Dokumentation bzw. ein Handbuch erforderlich ist.
- *Mitdenkend*: Die Suche sollte Vorschläge für Suchkriterien, nach denen gesucht werden könnte, liefern.
- *Motivierend*: Es sollte Spaß machen mit der Webanwendung zu arbeiten.
- *Schnell*: Die Antwortzeiten für die Suche müssen möglichst schnell sein.
- *Übersichtlich*: Das Suchinterface und die Ergebnisdarstellung müssen übersichtlich sein.
- *Umfassend*: Die Suche sollte auf einer möglichst umfassenden, Hersteller-übergreifenden Datenbank basieren.
- *Universell*: Die Suchanfrage soll universell einsetzbar sein; d. h., von überall über den Browser zugänglich sein.
- *Up-to-date*: Die der Suche zu Grunde liegende Datenbank sollte möglichst aktuell (up-to-date) sein.
- *Vielseitig*: Bei der Suche müssen möglichst viele Parameter berücksichtigt werden können und eine Hersteller-übergreifende Suche gewährleistet sein.

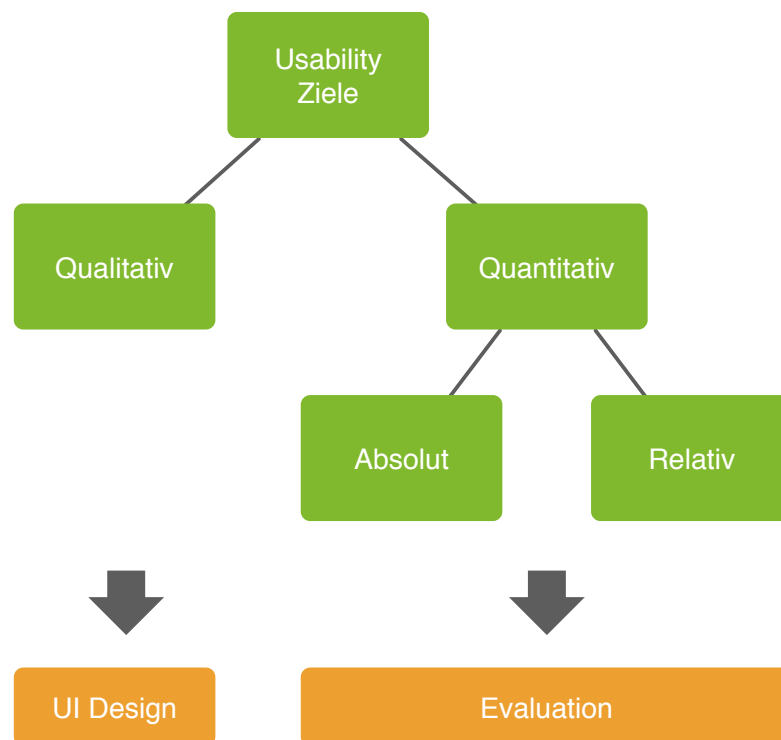
Die hier genannten Eigenschaften können gleichzeitig als Usability-Ziele für die Webanwendung verstanden werden.

## 3.3 Ableiten von Usability-Zielen

### 3.3.1 Zielsetzung

Die Usability-Ziele basieren auf den Ergebnissen der beiden vorangegangenen Schritte. Sie werden aus den Nutzerprofilen und der Aufgabenanalyse extrahiert. Usability-Ziele lenken das UI-Design als auch den Evaluierungsprozess. Sie dienen als Grundlage für

jede Designentscheidung und helfen dabei, den Designzyklus zu verkürzen und zu glätten. Außerdem werden sie als Akzeptanzkriterium während der Usability-Evaluation genutzt. Mit ihrer Hilfe kann im gesamten Projekt-Lebenszyklus Zeit gespart und ein besseres Design für die anvisierten Nutzergruppen gewährleistet werden. Einen Überblick über die verschiedenen Arten von Usability-Zielen zeigt Abbildung 3.1.



**Abbildung 3.1:** Überblick Usability-Ziele

Es gibt sowohl qualitative Ziele als auch quantitative Ziele. Die qualitativen Ziele sind allgemeine, nicht messbare Ziele, die das UI-Design lenken und vorantreiben. Sie beziehen sich auf das gesamte UI; d. h., auf das Suchinterface und die Visualisierung bzw. Ergebnisdarstellung. Die quantitativen Usability-Ziele dagegen, gelten als objektiv und messbar und werden während der späteren Evaluation überprüft. Diese kann man weiterhin in absolute und relative Ziele unterteilen. Absolute Ziele haben eine absolute Bewertung. Das kann z. B. die Zeit sein, die benötigt wird, um eine bestimmte Auf-

gabe zu erledigen oder die Anzahl und Art der Fehler bei einer bestimmten Aufgabe. Relative Ziele beziehen sich auf die Erfahrungen des Nutzers bzgl. des UI-Design des Produkts im Vergleich zu einem anderen Produkt. Beispielsweise sollte die Suche nach Aktoren mit der Webanwendung schneller bzw. präziser gehen als bei bisheriger Vorgehensweise oder im Vergleich zu einer alternativen Suchwebseite. Weiterhin kann man die quantitativen Ziele auch nach den Größen Effektivität und Präferenz/Zufriedenheit eines Nutzers unterteilen.

### 3.3.2 Vorgehensweise

Für das allgemeine Vorgehen bei dieser Aktivität werden

1. Usability-Ziele formuliert und
2. Usability-Ziele priorisiert.

#### 1. Usability-Ziele formulieren

Bei der Formulierung der Usability-Ziele, stehen hauptsächlich Ziele im Vordergrund, die für die Akzeptanz des zu entwickelnden Produktes wichtig sind. Diese sollten dementsprechend hoch priorisiert werden. Ziele, die nicht primär für die Akzeptanz notwendig sind, bekommen eine niedrige Priorität und sollten nur dann anvisiert werden, wenn sie nicht zu Lasten wichtigerer Ziele gehen und wenn sie nicht viel Zeit und keine hohe Kosten verursachen. Zu viele Ziele zu formulieren ist nicht praktikabel, da sie das Testen zu umfangreich und zeitaufwendig machen; d. h., die Anzahl der quantitativen Ziele für die spätere Evaluation sollte klein gehalten werden. Für die qualitativen Ziele, welche dem Designer helfen, Entscheidungen bzgl. des UI-Designs abzuwägen, gilt diese Regelung jedoch nicht. Bei der Entwicklung neuer Anwendungen ist es ratsam, den Fokus mehr auf allgemeine Usability-Ziele zu legen, wohingegen beim Redesign einer Anwendung, der Fokus eher auf spezifische, feature-orientierte Ziele liegt.



## 2. Usability-Ziele priorisieren

Bei der Priorisierung der Ziele muss unterschieden werden zwischen solchen, die für den Erfolg der Webanwendung fundamental wichtig sind; wichtigen Zielen, die nicht zu kosten- oder zeitintensiv sind und Zielen, die wünschenswert aber nicht wichtig sind. Dabei wurden folgende Priorisierungsklassen eingeführt:

**1** = erforderlich für das Release (Veröffentlichung)

**2** = wichtig, wenn nicht zu kosten- und zeitintensiv

**3** = wünschenswert, wenn einfach umsetzbar

Alle quantitativen Ziele mit hoher Priorität dienen später in der Evaluationsphase als Akzeptanzkriterium für die Veröffentlichung der Webanwendung.

### 3.3.3 Ergebnisse

Die aus den Nutzerprofilen und der Aufgabenanalyse abgeleiteten Usability-Ziele und ihre Priorität werden in der Tabelle 3.2 zusammengefasst. Diese spiegeln im wesentlichen die aus der Aufgabenanalyse erlangten Vorstellungen und Anforderungen der Nutzer bzgl. der Webanwendung wider.

## 3.4 Plattformbeschränkungen

Das Ziel dieser Phase ist es, das Potential und die Einschränkungen, welche die technologische Plattform (Betriebssystem, Browser, Internet etc.) für das UI-Design mitbringen, zu bestimmen und dokumentieren. Dies ist wichtig, da die Hardware- und Software-Plattformbeschränkungen die UI-Design Möglichkeiten stark einschränken können. Gerade bei Webseiten gibt es eine große Anzahl und Vielfalt von Hard- und Software-Plattformen, auf die bei der Entwicklung und Gestaltung geachtet werden muss. Um Informationen über die Hard- und Software-Plattform eines Produktes zu erhalten, ist es möglich, technische Dokumentationen einzusehen und Experten zu befragen.

Tabelle 3.2: Überblick Usability-Ziele

Usability Ziele	Priorität
<b>qualitative Ziele</b>	
<i>Eindeutig</i> : Die Suchergebnisse sollten möglichst eindeutig und treffsicher sein.	1
<i>Erweiterbar</i> : Die Funktionalitäten der Aktoren-Suche sollten erweiterbar sein.	1
<i>Hierarchisch</i> : Die Eingrenzung der optimalen Lösung sollte trichterförmig geschehen.	1
<i>Interaktiv</i> : Die Suche und Visualisierung soll interaktiv bedienbar sein.	1
<i>Intuitiv</i> : Die Bedienung der Webanwendung muss intuitiv erfolgen können. Dabei ist es wichtig, dass die GUI-Elemente verständlich und das Verhalten der Webanwendung den Erwartungen des Nutzers und seinem mentalen Modell entspricht.	1
<i>Konsistent</i> : Die Gestaltung des gesamten UI sollte konsistent sein.	2
<i>Unterstützend</i> : Die Suche sollte Vorschläge für Suchkriterien, nach denen gesucht werden könnte, vorschlagen.	3
<i>Motivieren</i> : Es sollte Spaß machen, mit der Webanwendung zu arbeiten.	1
<i>Schnell</i> : Die Antwortzeiten für die Suche müssen möglichst schnell sein.	2
<i>Übersichtlich</i> : Das Suchinterface und die Ergebnisdarstellung müssen übersichtlich sein.	1
<i>Umfassend</i> : Die Suche sollte auf einer möglichst umfassenden, Hersteller-übergreifenden Datenbank basieren.	1
<i>Universell</i> : Die Suchanfrage soll universell einsetzbar sein; d. h., von überall über den Browser zugänglich sein.	1
<i>Up-to-date</i> : Die der Suche zu Grunde liegende Datenbank sollte möglichst aktuell (up-to-date) sein.	2
<i>Vielseitig</i> : Bei der Suche müssen möglichst viele Parameter berücksichtigt werden können und eine herstellerübergreifende Suche gewährleistet sein.	1
<b>quantitative Ziele</b>	
Die Aktorensuche mit der Webanwendung muss schneller sein, als mit der bisherigen Vorgehensweise.	1
80% der Nutzer bewerten die Webanwendung als übersichtlich.	1
80% der Nutzer sind mit der Webanwendung zufrieden.	1

Die Plattformen der Nutzer können in den folgenden Punkten variieren:

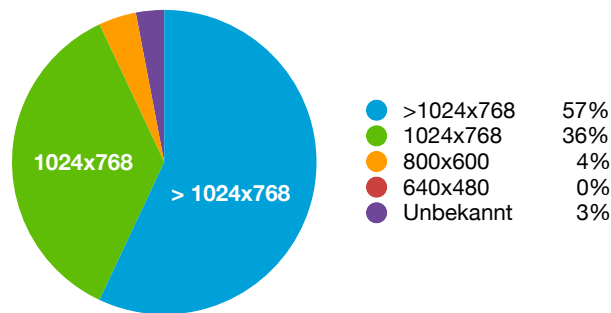
- Bildschirmgröße und Auflösung,
- Geschwindigkeit der Internetverbindung,
- Browser-Fähigkeiten.

Die Expertenbefragung zu den Nutzerprofilen hat gezeigt, dass die Hardwarevoraussetzungen bei den unterschiedlichen Nutzergruppen fast identisch sind. Alle Nutzertypen verfügen über eine sehr schnelle Internetverbindung und sehr gute Rechenleistung. Hauptsächlich bei der Größe des verwendeten Monitors; d. h., bei der Auflösung, gibt es Unterschiede. Der Grund dafür ist, dass sowohl Desktop Rechner als auch Notebooks bei der Arbeit und entsprechend bei der Suche nach Akteuren zum Einsatz kommen können.

Bei der Entwicklung für eine Webanwendung, ist es wichtig zu wissen, welche UI-Design Möglichkeiten sich anbieten, auch wenn sie evtl. nicht einfach realisierbar sind und welche absolut unmöglich sind [May99].

### **Bildschirmgröße und Auflösung**

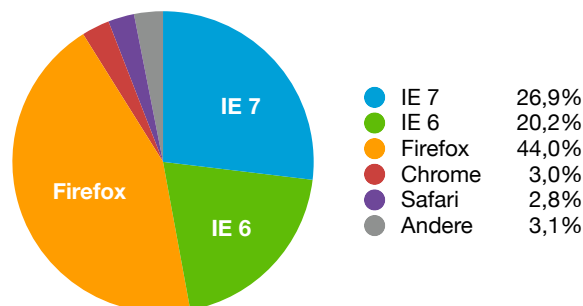
Die Befragung ergab, dass die Nutzer Bildschirmauflösungen von 1440 x 900 Pixel und darüber hinaus verwenden. Lediglich in der Projektleitung, wo auch Notebooks zum Einsatz kommen, gibt es noch Auflösungen mit 1024 x 768 Pixeln. Dieses Ergebnis zeigt sich auch in der Studie von [www.w3schools.com](http://www.w3schools.com) [w3s09]. Diese besagt, dass 57% der Internet-Nutzer eine Auflösung verwenden, die größer als 1024 x 768 Pixel ist und immer noch 36% der Internet-Nutzer mit einer Auflösung von 1024 x 768 Pixeln auskommen müssen (siehe Abbildung 3.2). Die Webanwendung sollte demzufolge auch in dieser Auflösung gut dargestellt werden können und bei höheren Auflösungen sich flexibel an die Größe des Browserfensters anpassen können.



**Abbildung 3.2:** Bildschirmauflösungs-Statistiken [w3s09]

### Browser Fähigkeiten

Nach der gleichen Studie [w3s08] sehen die aktuellen Anteile im Browser-Markt wie in Abbildung 3.3 dargestellt aus. Solche Statistiken sollten allerdings im Allgemeinen nur



**Abbildung 3.3:** Browser-Statistiken [w3s08]

als Orientierung dienen, da sie oft irreführend sein können. Globale Browser-Statistiken müssen nichts über die Browser-Statistiken für die zu entwickelnde Webseite aussagen. Jede Webseite spricht ein unterschiedliches Publikum an. Bestimmte Webseiten werden von Personen besucht, die Profis im Umgang mit dem Internet sind und über High-End-Hard- und Softwareausstattungen verfügen. Andere wiederum zielen eher auf Hobby-Internet-Surfer ab, die nur über eine Low-End-Ausrüstung verfügen und zum Beispiel mit dem vom Betriebssystem vorinstallierten Browser arbeiten. Aus der

Aufgabenanalyse und den dafür geführten Interviews geht hervor, dass vorwiegend die Browser *Firefox* und *IE 7* (oder neuer) eingesetzt werden. Auch bei Nutzertypen handelt es sich mehr um den Profi-Nutzer, da ihre Arbeit ein ständiges Recherchieren und Arbeiten im Internet voraussetzt, und sie über gute Hardwarevoraussetzungen und eine schnelle Internetverbindung verfügen.

## 3.5 Designprinzipien

Diese Aktivität kann, wie die Plattformbeschränkungen, zu jeder Zeit der Anforderungsanalyse im UEL durchgeführt werden. Ziel ist es, allgemeine Designprinzipien und Guidelines, welche auf das UI-Design der zu entwickelnden Webanwendung anwendbar sind, zu identifizieren. Dazu wurde eine Literaturrecherche zu verfügbaren Guidelines und Styleguides durchgeführt. Die Designprinzipien dienen als Leitfaden während des gesamten Designprozesses. In der Praxis haben sich eine ganze Reihe von Regeln, Standards, Vorschriften etc. herausgebildet, welche sich die zum Thema Gestalten von Webanwendungen und deren Usability beschäftigen. An dieser Stelle sollen drei Regeln kurz näher beschrieben werden.

### Grundsätze der Dialoggestaltung (DIN EN ISO 9241-10)

Wenn es um die Nutzerfreundlichkeit bezüglich der Interaktion vom Nutzer mit dem System (Webanwendung) geht, können folgende Grundsätze zur Dialoggestaltung angewendet werden [DIN03]:

- *Aufgabenangemessenheit*: Das System unterstützt die Arbeitsaufgaben seiner Benutzer effektiv und effizient.
- *Selbstbeschreibungsfähigkeit*: Das System enthält Erläuterungen und ist ausreichend verständlich.
- *Steuerbarkeit*: Der Benutzer muss in der Lage sein, die Richtung und Geschwindigkeit des Dialogsystems zu beeinflussen, bis das Ziel erreicht ist.

- *Erwartungskonformität*: Das System muss konsistent sein und die Erwartungen, Eigenschaften und Gewohnheiten des Nutzers unterstützen.
- *Fehlertoleranz*: Fehlerhafte Eingaben müssen mit minimalem Korrekturaufwand durch den Benutzer zum beabsichtigten Arbeitsergebnis führen.
- *Individualisierbarkeit*: Das System kann an die individuellen Bedürfnisse des Nutzers und den Erfordernissen der Arbeitsaufgabe angepasst werden.
- *Lernförderlichkeit*: Der Benutzer muss beim Erlernen des Systems bzw. neuer Funktionen unterstützt und angeleitet werden.

### Spezielle Gestaltungsgrundsätze für Multimedia

Für die Gestaltung von Multimedia-Schnittstellen sollten zusätzlich zu den allgemeinen Grundsätzen der Dialoggestaltung (ISO 9241-10), folgende vier speziellen Gestaltungsgrundsätze berücksichtigt werden (siehe [DIN03]):

- *Eignung für das Kommunikationsziel*  
“Eine Multimedia-Anwendung ist für das Kommunikationsziel geeignet, wenn sie so gestaltet ist, dass sie sowohl den Zielen des (der) Anbieter(s) der zu übermittelnden Information als auch dem Ziel oder der Aufgabe der Benutzer oder Empfänger dieser Information entspricht.”
- *Eignung für Wahrnehmung und Verständnis*  
“Eine Multimedia-Anwendung ist für Wahrnehmung und Verständnis geeignet, wenn sie so gestaltet ist, dass die zu übermittelnde Information leicht erfasst und verstanden werden kann.”  
Um dies zu gewährleisten, sollten die Eigenschaften (ISO 9241-12) Erkennbarkeit, Unterscheidbarkeit, Klarheit, Lesbarkeit, Konsistenz, Kompaktheit und Verständlichkeit im Design umgesetzt werden.
- *Eignung für Exploration*  
“Eine Multimedia-Anwendung ist für die Exploration geeignet, wenn sie so gestaltet ist, dass der Benutzer eine relevante oder interessante Information mit wenig

oder keinem Vorwissen in Bezug auf Art, Umfang oder Struktur der Information oder der verfügbaren Funktionalität der Anwendung finden kann.”

- *Eignung für Benutzungsmotivation*

“Falls es für die Arbeitsaufgabe angebracht ist, sollte eine Multimedia-Anwendung so gestaltet sein, dass sie für den Benutzer anregend ist, d. h. dass sie die Aufmerksamkeit des Benutzers auf sich zieht und ihn dazu motiviert, mit ihr zu interagieren.”

### “Acht Goldene Regeln” des Schnittstellen-Designs

Der Experte im Bereich Mensch-Computer-Interaktion, *Ben Shneiderman* (vgl. [Shn02], [SP04]), empfiehlt, als Grundlage für Web-Usability seine “*Acht Goldene Regeln*” des Schnittstellendesigns zu beachten. Danach soll eine Anwendung immer:

- nach Konsistenz streben;
- regelmäßigen Benutzern die Verwendung von Shortcuts ermöglichen;
- in sich geschlossene Dialoge benutzen;
- Fehler, die bei der Benutzung auftreten können, vermeiden oder leicht kontrollierbar machen;
- leichte Umkehr von Aktionen ermöglichen;
- das interne Kontrollbedürfnis der Nutzer unterstützen und
- die Belastung des Kurzzeitgedächtnisses der Benutzer reduzieren.

Eine ausführlichere Beschreibung dieser Regeln ist in [Shn02, S.100f] nachzulesen.

# Kapitel 4

## Design

Die Ergebnisse der Anforderungsanalyse (Kapitel 3) fließen direkt in den Designprozess mit ein. Bevor jedoch das Design weiter ausgeführt wird, erfolgt ein Überarbeiten des aktuellen Arbeitsmodells (Vorgehensmodells) des Nutzers. Dabei sollte dieses Modell nur soweit geändert werden, dass eine bessere Automatisierung einzelner Arbeitsschritte möglich ist; d. h., ein besserer Workflow zustande kommt [May99]. Dieser Schritt kann in einem neuen Modell, welches die Funktionalität und die Struktur des zu erstellenden Produktes beschreibt, dokumentiert und später validiert werden. Bei Projekten mit geringen Ressourcen ist es möglich, die Dokumentation und Validierung zu verkürzen oder sogar ganz wegzulassen (vgl. [May99, S.182]). Jedoch sollten trotzdem zweckdienliche Notizen gemacht werden, um sie im anschließenden Conceptual Model Design zu berücksichtigen.

Die Designphase ist in drei Ebenen eingeteilt, welche sich durch einen Top-Down Ansatz im UI-Design auszeichnen. In der ersten Ebene, dem *Conceptual-Model-Design*, wird das konzeptionelle Design festgelegt. Dabei geht es um das High-Level Design; d. h., es wird untersucht, wie ist die Anwendung aufgebaut, was ist hauptsächlich zu sehen und wie bewegt sich der Nutzer zwischen den verschiedenen Bereichen. Hier werden erste Mock-Ups<sup>1</sup> erstellt. Danach wird der aus dem vorherigen Schritt entworfene Prototyp zu einem *Screen-Design* weiterentwickelt; d. h., es wird ein detaillierter

---

<sup>1</sup>bezeichnet einen rudimentären Prototypen bzw. ein Vorführmodell einer Benutzeroberfläche



Prototyp für die Oberfläche entworfen. Dieser soll die festgelegten Designstandards verdeutlichen. In der dritten Ebene wird das komplette, *detaillierte UI-Design* fertiggestellt. Grundlage dafür bilden die vorangegangenen Schritte.

Für relativ einfache Webseiten und Anwendungen ist es zulässig, die drei Ebenen zu einer zusammenzufassen [May99, S. 249] und erst danach die Evaluation durchzuführen. Für Webseiten von mittlerer Komplexität ist es allerdings besser, die erste Phase des Designs, das Conceptual-Model-Design (High-Level Design), ganz normal auszuführen und mit Hilfe einer *Heuristischen Evaluation*<sup>2</sup> [Nie93, S. 155] oder eines *Reviews* durch einen Usability Experten zu überprüfen. Die zwei folgenden Design Phasen (Screen Design und detailliertes UI-Design) sollten als ein iterativer Design- und Evaluation-Prozess zusammen durchgeführt werden. Nachdem das detaillierte UI fertig entworfen ist und es einen funktionsfähigen Prototypen gibt, können erste Usability-Tests durchgeführt werden.

### Vorgehensweise

Das hier gewählte Vorgehen, welches in Abbildung 2.3 verdeutlicht wird, sieht wie folgt aus: Zuerst werden die Problematik bei der Suche und das Vorgehensmodell des Nutzers beschrieben (Work-Reengineering). Dann wird das Conceptual-Model-Design entwickelt und mit Hilfe der heuristischen Evaluation überprüft. Anschließend wird das Screen-Design und detaillierte UI-Design in einem Schritt angefertigt. Ist dieses abgeschlossen, erfolgen Usability-Test, welche den entstandenen Prototypen hinsichtlich der quantitativen Usability-Ziele überprüfen.

## 4.1 Work-Reengineering

Um ein verbessertes Vorgehensmodell des Nutzers zu beschreiben, muss noch einmal kurz auf die Besonderheiten bei der Suche nach multidimensionalen (hypervarianten) Daten und der Auswahl eines Ergebnisses aus vielen eingegangen werden. Bei der Su-

---

<sup>2</sup>ist eine systematische Betrachtung des UI-Designs nach Usability-Kriterien

che und Auswahl von Aktoren ist die Problematik die folgende: Es gibt eine riesige Anzahl von Aktoren, bei der jeder Aktor durch bestimmte Parameter mit verschiedenen Werten (Parametern) gekennzeichnet ist. Der Ingenieur ist bestrebt, aus dieser großen Menge an Aktoren, jenen zu finden, der für seine Problemstellung am besten geeignet ist oder eine kleine Anzahl von Aktoren zu finden, welche man etwas detaillierter betrachten und vergleichen möchte. Die Aufgabe “Einen aus Vielen” zu wählen, unter Berücksichtigung verschiedener Parameter, wie sie bei der Aktorsuche auftritt, ist heutzutage in vielen Bereichen weit verbreitet (z.B.: Investitionsentscheidungen, Auto-Kauf, Notebook-Kauf etc.). Bei normalen Datenbankabfragen kann es oft passieren, dass sie zu keinen Treffern führen. Es ist dann schwierig herauszufinden, an welchem Kriterium die Suche gescheitert ist. Wenn die Suche angefangen wird, ist man sich oft noch nicht sicher, was genau man sucht [Rei05]; d. h., die Suche und Auswahl hängen von den zur Verfügung stehenden Möglichkeiten und ihren Eigenschaften ab. Bei der Aktorsuche ist das genauso. Erst wenn man weiß, welche Möglichkeiten es insgesamt gibt, kann man die Aktoren bezüglich den gewünschten Parametern besser untereinander vergleichen. Dies hilft dann beim Bewerten der gefunden Ergebnisse und beim Inspirierenlassen für weitere Einschränkungen der Suche. Folgendes Beispiel soll diesen Sachverhalt besser verdeutlichen. Wenn der Ingenieur einen Aktor (Aktorklasse) während seiner Recherche entdeckt, welcher zwar sein Preislimit etwas übersteigt, aber viel bessere Eigenschaften besitzt als vorher angenommen, würde er evtl. sein vorher angegebenes Preislimit ausdehnen. Normale Anfragetechniken, wie man sie aus Datenbanksystemen oder Produktuchmaschinen bisher kennt, unterstützen den Nutzer nicht, den Informationsraum zu erforschen. Sie dienen nur der gezielten Suche. Hier wäre es wahrscheinlich nicht möglich, diese neue Aktorklasse bei der Recherche zu entdecken und für die vom Ingenieur gegebene Problemstellung als Lösung mit in Betracht zu ziehen.

Die hier zu entwickelnde Suche und Visualisierung muss dementsprechend sowohl die gezielte Exploration (Suche) als auch die freie Exploration (Browsen) der Daten ermöglichen. Dabei muss das Hin- und Herwechseln zwischen beiden Formen für den Nutzer jederzeit fließend möglich sein (siehe [Rei05, S. 382]). Nach [Shn02] ist das Mantra zur Erfassung visueller Informationen: “Zuerst Überblick, Zoomen und Filtern, dann die

Details nach Bedarf“. Diese Vorgehensweise soll bei der Gestaltung des Suchinterfaces und der Visualisierung berücksichtigt werden und dient als grobes Vorgehensmodell des Nutzers. Den Nutzern sollte außerdem eine Art Spielwiese (Playground) angeboten werden, wo sie die Parameter der Aktoren frei manipulieren und damit herumspielen können (vgl. [Spe07, S. 199]). Dies hilft beim Erforschen des gegebenen Datenbestandes und lässt Muster, Lücken und Ausreißer besser erkennen.

## 4.2 Conceptual-Model-Design

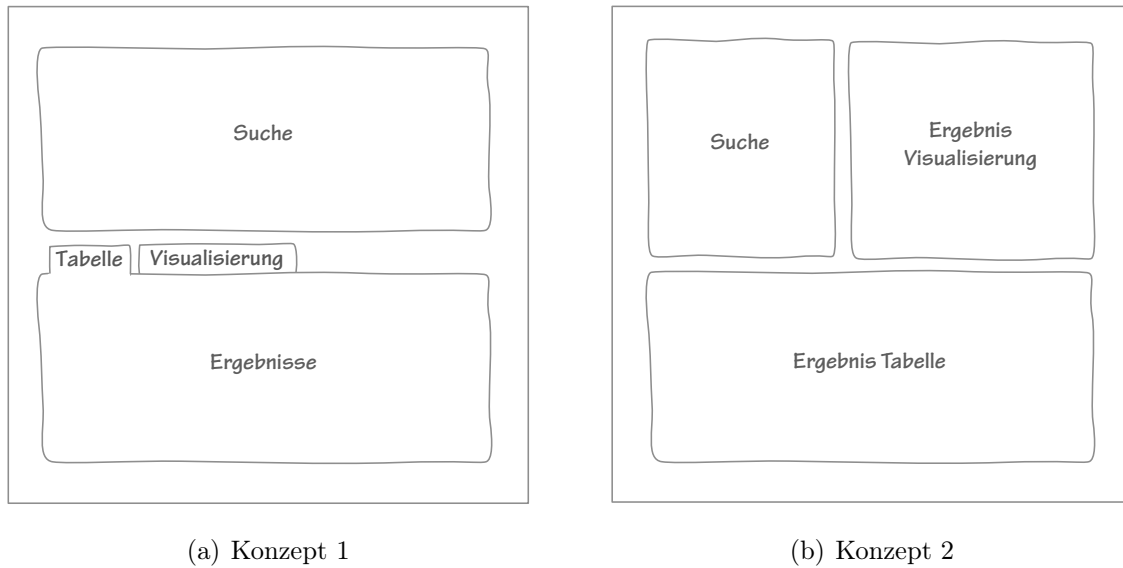
### 4.2.1 Zielsetzung

In dieser Phase wird das High-Level Design für die Webanwendung aufgestellt und definiert. Dabei müssen die Ergebnisse aus der Anforderungsanalyse (Kapitel 3) und dem Work-Reengineering (Kapitel 4.1) umgesetzt werden. Im High-Level Design wird der grundsätzliche Aufbau der Anwendung (Screen-Layout) und die Aufteilung sowie Struktur der Information festgelegt. Folgende Punkte spielen dabei eine wichtige Rolle. Die Haupt-Anzeigebereiche müssen identifiziert und die Navigationswege zwischen den Bereichen sowie die Interaktionsprinzipien sollten beschrieben werden.

### 4.2.2 Vorgehensweise

Es wurden zwei Konzepte (siehe Abbildung 4.1) entworfen und ihre Vor- und Nachteile mittels einer Heuristischen Evaluation untersucht. Beim Erstellen der beiden Konzepte flossen die Ergebnisse aus der Anforderungsanalyse und dem Work-Reengineering mit ein. Beide Konzepte müssen sowohl das Suchen als auch das Browsen nach Aktoren unterstützen.

Das Layout der Webanwendung unterteilt sich in die Bereiche: Suche, Ergebnisvisualisierung und Ergebnistabelle.



**Abbildung 4.1:** Conceptual-Model-Design des UI

### Suche

Der Bereich Suche dient zum Formulieren der Suchanfrage. Dabei können die einzelnen Kriterien für die jeweiligen Parameter angegeben werden. In diesem Bereich befindet sich außerdem der Such-Button zum manuellen Anstossen der Suche und eine Anzeige für die Ergebnisanzahl. Nachdem eine erste Suche erfolgt ist, dient dieser Bereich zur weiteren Einschränkung der Suche. Wichtig ist dabei, dass das Suchinterface immer sichtbar und zugänglich ist. Nutzer sollen die Möglichkeit haben, die Suche zu jeden Zeitpunkt beeinflussen zu können. Nach [Shn02, S. 586] haben empirische Untersuchungen belegt, dass Nutzer zufriedener und besser arbeiten, wenn sie die Suche sehen und kontrollieren können.

### Ergebnisvisualisierung

Dieser Bereich beinhaltet die grafische Darstellung der Ergebnisse und soll primär dazu genutzt werden, bestimmte Parameter und ihre Abhängigkeiten grafisch darzustellen. Sie soll vor allem einen Überblick über den Datenbestand bzw. die verfügbaren Ergebnisse liefern. Es kann z. B. genutzt werden, um Ausreißer besser zu erkennen und um Anregungen für weitere Einschränkungen bei der Suche zu erhalten.

### Ergebnistabelle

Die Ergebnistabelle soll eine detaillierte Darstellung der vorhandenen Ergebnisse liefern und das Vergleichen, Sortieren sowie weitere Aktionen ermöglichen. Weitere Aktionen können z. B. erneutes Filtern der Ergebnisse sein bzw. Verlinken zu der Hersteller-Webseite, um den Aktor zu bestellen oder Einsicht in Datenblätter vom Hersteller zu erhalten. Weil die Tabelle alle wichtigen Parameter der Aktoren darstellen soll, ist es notwendig, dass dieser Bereich über die ganze Breite des Browserfensters reicht.

Die hier beschriebene Unterteilung ermöglicht es, dass zu jedem Zeitpunkt eine gezielte Suche und freie Exploration möglich ist. Außerdem hat der Nutzer die Möglichkeit, jederzeit die Details der vorliegenden Ergebnisse in der Ergebnistabelle zu betrachten.

#### 4.2.3 Ergebnis

Wichtig ist, dass das Suchinterface und die Visualisierung für den Nutzer gleichzeitig sichtbar ist. Es muss möglich sein, die Suchkriterien beeinflussen zu können und gleichzeitig die Auswirkung auf die Ergebnismenge zu betrachten, ohne das dafür gescrollt werden muss. In Variante 1 ist dies nicht möglich, da die Ergebnisse im Bereich darunter dargestellt werden und normale Monitore ein Querformat benutzen. Das zweite Layout hat hier klare Vorteile. Es lassen sich leichter Änderungen in der Suche vornehmen und gleichzeitig ihre Auswirkungen auf die Ergebnisse in der Visualisierung erkennen, weil diese beiden Bereiche nebeneinander angeordnet sind und somit gleichzeitig sichtbar sind. Das erste Konzept stellt sich außerdem beim Umschalten zwischen den Ansichten Visualisierung, für den Überblick und die Tabelle, für die Details als zu aufwendig heraus. Das zweite Konzept ermöglicht dies einfacher, und bei größeren Monitoren entfällt außerdem das scrollen, da alle drei Bereiche auf eine Seite passen. Aus den hier beschriebenen Gründen wurde sich für das zweite Konzept entschieden, welches in der nachfolgenden Designphase weiter verfeinert wird.

## 4.3 Screen-Design und detailliertes UI-Design

Nachdem das konzeptuelle High-Level Design festgelegt wurde, kann nun das Design weiter verfeinert werden. Im ersten Teil soll zunächst die Verfeinerung des Suchinterface beschrieben werden. Im zweiten Teil wird dann die Ergebnisdarstellung (Visualisierung) betrachtet.

### 4.3.1 Suchinterface

Fundamental wichtig für eine effiziente und effektive Nutzung der Webanwendung zur Aktorsuche ist die Gestaltung des UI. Dieses hat direkten Einfluss auf die Usability der Webanwendung und muss demzufolge für Nutzer, die selten oder zum ersten Mal die Anwendung bedienen (Anfänger) als auch Nutzer die häufig damit arbeiten (Profi), gut bedienbar sein.

Bei der Gestaltung des Suchinterfaces gab es bestimmte Anforderungen (siehe Kapitel 3.2 und 3.3), die es zu berücksichtigen galt. Das Interface musste klar strukturiert sein, um die Übersichtlichkeit zu gewährleisten. Bei der Suche musste eine schrittweise Einschränkung der Ergebnisse möglich sein. Alle Parameter sollten bei der Suche verfügbar sein, aber deren Eingabe nicht zwingend verlangt werden. Außerdem sollte es eine Anzeige für die aktuelle Ergebnisanzahl geben.

Das Suchinterface lässt sich grob in drei Bereiche unterteilen: Auswahl der Bewegungsform, Eingabe der Suchkriterien und Anzeige für die Ergebnisanzahl.

Dabei wurden zwei Varianten ausgearbeitet, um das Suchinterface unter Berücksichtigung der Usability-Ziele umzusetzen (siehe Abbildung 4.2 und 4.3). Beide Varianten unterscheiden sich nur bei der Eingabe der Suchkriterien.

- *Variante 1*: Gleichzeitige Anzeige aller Parameter (die Werte enthalten).
- *Variante 2*: Anzeige ausschließlich der Parameter, nach denen gesucht wird.

Bei beiden Varianten muss zu Anfang die Bewegungsform des Aktors (rotatorisch oder transitorisch), nach der gesucht werden soll, bestimmt werden. In Abhängigkeit von der

**Bewegungsform** ☒ Rotatorisch ☐ Translatorisch

---

Typ

Hersteller

Bewegungssinn

Bewegungsverlauf

---

Drehzahl   und  1/s

Wirkungsgrad

Leistung   W

Breite generalisiert

Länge

Masse   kg

Volumen   und  m<sup>3</sup>

Dichte

**6 Ergebnisse**

Abbildung 4.2: Variante 1: Gleichzeitige Anzeige aller Parameter

**Bewegungsform** ☒ Rotatorisch ☐ Translatorisch

---

☒ Typ

☒ Volumen   und  m<sup>3</sup>   **6 Ergebnisse**

☒ Drehzahl   und  1/s   **6 Ergebnisse**

☒ Leistung   W   **6 Ergebnisse**

☒ Masse   kg   **6 Ergebnisse**

Abbildung 4.3: Variante 2: Anzeige ausschließlich der gesuchten Parameter

getroffenen Wahl werden die verfügbaren Parameter bereitgestellt. Bei der Variante 1 sind die Parameter (Kriterien) alle gleichzeitig sichtbar, und der Nutzer kann die Parameter, die ihn interessieren, verändern. Bei Variante 2 kann der Nutzer neue Kriterien beliebig hinzufügen und den Parameter, nach den er suchen will, einzeln auswählen. Es wird nur nach den Kriterien gesucht, bei denen eine Änderung vorgenommen wurde. Alle anderen werden als beliebig behandelt und schränken somit den Datenbereich nicht weiter ein. Ist ein Kriterium angegeben, wird automatisch berechnet, wie viele Ergebnisse die Kriterien erfüllen bzw. wie viel Ergebnisse der Suchalgorithmus erfasst hat. Die Anzahl der Ergebnisse wird dann im unteren Bereich des Suchinterfaces prominent dargestellt.

Variante 2 hat den Vorteil, dass nur die Kriterien sichtbar werden, nach denen gesucht wurde und die dementsprechend wirklich relevant sind. Dies trägt bei vielen vorhandenen Parametern zu deutlich mehr Übersichtlichkeit bei und benötigt weniger Platz am Bildschirm. Diese Variante hat jedoch auch einen Nachteil. Wenn man noch nicht genau weiß, wonach man genau suchen will (Browsen) ist es besser, wenn man alle Kriterien auf einen Blick sichtbar vor sich hat; vorausgesetzt, es sind nicht zu viele Kriterien. Dadurch hat man einen schnelleren Überblick wonach gesucht werden könnte und es ist einfacher, die angezeigten Kriterien beliebig zu verändern. Die Variante 2 weist ein weiteres Interface Problem auf. Weil durch das beliebige Hinzufügen neuer Kriterien mehr Bedienelemente nötig sind, wird das Suchinterface komplexer und es kann bei anfänglicher Benutzung zu Unklarheiten über die Funktion der einzelnen Bedienelemente kommen. Zum Beispiel können die Plus- und Minus- Buttons<sup>3</sup> als Gewichtung der einzelnen Kriterien verstanden werden und nicht als Möglichkeit, ein neues Kriterium hinzuzufügen.

Bei der zu erstellenden Webanwendung handelt es sich um die erste Version einer Aktorsuche im Internet. Diese dient als Prototyp und soll die Möglichkeiten der Aktorsuche und den Nutzen geeigneter Visualisierungsmöglichkeiten verdeutlichen. Dadurch, dass die Variante 1 im Rahmen der Arbeit leichter umzusetzen ist und sie die

---

<sup>3</sup>Bedienelement in grafischen Benutzeroberflächen, auch Taste oder Knopf genannt



Usability-Ziele (Kaptitel 3.3) dennoch berücksichtigt, wird sie für das Suchinterface umgesetzt.

Wenn in Zukunft immer mehr Parameter für die Aktoren hinzukommen und somit die Liste der Kriterien immer länger wird, ist es sinnvoller, die Variante 2, wo Kriterien beliebig hinzugefügt werden können, zu realisieren. Dabei könnte auch ein Schieberegler für die Bereichssuche (größer, kleiner, im Bereich) zum Einsatz kommen (siehe Abbildung 4.4).

The screenshot shows a search interface titled "Bewegungsform" with two radio buttons: "Rotatorisch" (selected) and "Translatorisch". Below this, there are five filter criteria, each with a checked checkbox, a dropdown menu, a range selector, and a unit:

- Typ**: Dropdown menu with "ist nicht" and "EC-Motor". Range selector from 0 to 100 with markers at 25 and 75. Unit: m<sup>3</sup>.
- Volumen**: Range selector from 0 to 100 with markers at 25 and 100. Unit: 1/s.
- Drehzahl**: Range selector from 0 to 100 with markers at 25 and 100. Unit: W.
- Leistung**: Range selector from 0 to 100 with markers at 0 and 25. Unit: kg.
- Masse**: Range selector from 0 to 100 with markers at 0 and 25. Unit: kg.

At the bottom left is a "Suchen" button. At the bottom right, it says "6 Ergebnisse" in orange text.

**Abbildung 4.4:** Variante 2 mit Schieberegler

Usability-Tests müssten die Tauglichkeit jedoch erst unter Beweis stellen. Bei dem Interview in der Aufgabenanalyse hat sich herausgestellt, dass Schieberegler für die Angabe von Kenngrößen zu ungenau sein könnten. Eine Kombination von Schieberegler und der Möglichkeit, den Wert auch exakt eingeben zu können, könnte dabei eine Lösung sein. Generell ist zu sagen, dass im Suchinterface der Aktorsuche durchaus noch Verbesserungspotential vorhanden ist.

### 4.3.2 Ergebnisvisualisierung

Auf Grund der Menge und Komplexität der darzustellenden Daten, spielt eine geeignete visuelle Repräsentation eine große Rolle. Dabei ist die Reduktion der Daten auf eine zu bewältigende Informationsmenge enorm wichtig. Sie soll dem Nutzer bei der Analyse und dem Verständnis der dargestellten Daten helfen und die Kommunikation des visualisierten Inhalts vereinfachen (vgl. [SM00]).

Nach [SM00] muss eine Visualisierung expressiv, effektiv und angemessen sein:

- *Expressivität*: Die Daten dürfen durch die Visualisierung nicht verfälscht oder verzerrt werden.
- *Effektivität*: Die Visualisierung muss dem Nutzer die Information intuitiv vermitteln.
- *Angemessenheit*: Die Kosten für die Durchführung einer Visualisierung müssen angemessen sein<sup>4</sup>.

Visualisierungen gab es schon weit vor der Benutzung von Computern [Spe07] und finden heutzutage in vielen Bereichen Anwendung. Jedoch stoßen sie bei Web-, Produkt- sowie Katalog- und Online-Hilfe-Suchsystemen heutzutage auf wenig Akzeptanz und können sich kaum durchsetzen [Dem07]. Dort kommen weiterhin herkömmliche Konzepte (listenbasierte Darstellung der Ergebnismenge) zum Einsatz. Ein Grund dafür könnte sein, dass viele Visualisierungen einfach nicht alltagstauglich sind, da sie zu komplex und zu schlecht zu verstehen sind; d. h., die bisher verwendeten Ansätze weisen keinen wirklichen Mehrnutzen oder Effizienzgewinn bei der Suche auf [Naj03]. Visuelle Suchsysteme benötigen einen gewissen Trainingsaufwand, bis sie effektiv und effizient von Benutzern eingesetzt werden können [SVM99]. Ein bekanntes Problem bei der Entwicklung von visuellen Suchsystemen ist die schlechte Benutzerzentrierung [Rei05, S. 389ff]. Für die Aktorsuche muss also eine Visualisierung gefunden werden, die leicht und intuitiv zu verstehen ist und die der Nutzer evtl. schon aus anderen

---

<sup>4</sup>Angemessenheit beschreibt nicht die Qualität einer resultierenden Visualisierung

Bereichen kennt (Effektivität). Außerdem muss sie die Daten unverfälscht wiedergeben, um eine Fehlinterpretationen mit falschen Schlussfolgerungen des Betrachters zu reduzieren (Expressivität).

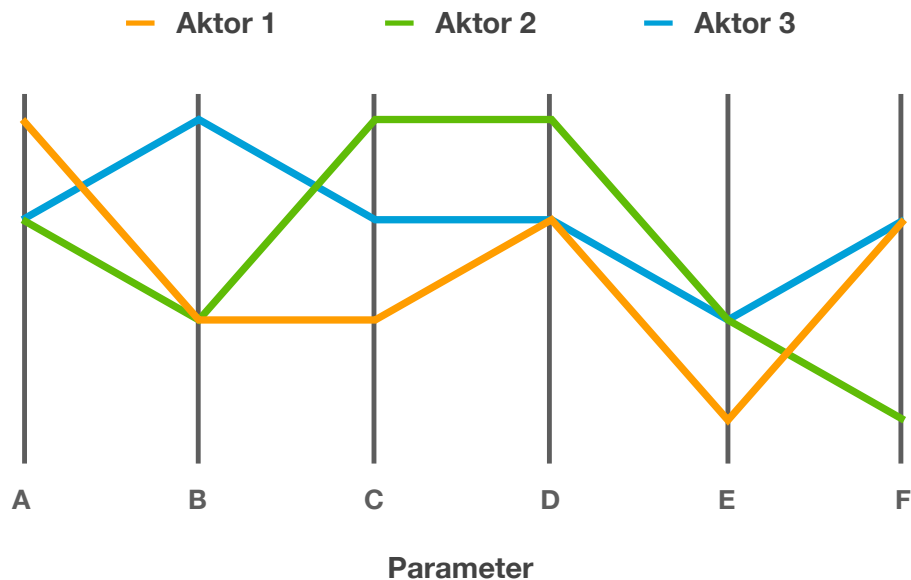
#### 4.3.2.1 Visualisierungsmöglichkeiten

Es gibt unzählige Möglichkeiten, wie Daten visuell dargestellt bzw. kodiert werden können, sodass an dieser Stelle bei weitem nicht alle Möglichkeiten und Techniken aufgelistet werden können. Es sollen somit nur Visualisierungsmöglichkeiten aufgelistet werden, die zur Darstellung von hypervarianten (multivarianten) Daten geeignet sind (siehe [Spe07]). Dabei ist es wichtig, dass die Visualisierungen sowohl *Attribut-* als auch *Objekt-Sichtbarkeit* gewährleisten. Von Objekt-Sichtbarkeit spricht man, wenn ein Nutzer die unterschiedlichen Attribute eines Objektes sehen kann und wie die verschiedenen Objekte zueinander in Beziehung stehen. Von Attribut-Sichtbarkeit spricht man, wenn ein Nutzer die Verteilung der verschiedenen Objekte bzgl. eines Attributes sehen kann. Zum Beispiel um herauszubekommen, ob es bei einem Attribut in einer bestimmten Region zu Clustern kommt oder um Ausreißer und Muster zu erkennen. Möchte man mehr über State-of-the-Art Methoden zur Visualisierung von Daten oder Informationsvisualisierung erfahren, empfiehlt es sich, [Lud04] und [Spe07] anzuschauen. Die folgenden Visualisierungsmöglichkeiten werden anschließend näher vorgestellt:

- *Parallel Coordinate Plot*,
- *Star Plot*,
- *Scatter Plot* (z. B. *FilmFinder*),
- *Bubble Chart*,
- *Bar Chart*,
- *Linked Histograms* (z. B. *Attribute Explorer*) und
- *Table Lens*.

### Parallel Coordinate Plot

Der *Parallel Coordinate Plot* ist eine der bekanntesten Techniken zur Darstellung von hypervarianten Daten und gehört zu der Kategorie der *Coordinate Plots* [Spe07, S. 56]. Für jede darzustellende Dimension (Parameter) wird eine eigene Achse verwendet. Für jeden Akteur, der durch verschiedene Parameter gekennzeichnet ist, wird der entsprechende Werte der einzelnen Parameter an den jeweiligen Achsen abgetragen. Diese werden mit einer Linie verbunden, um die Zusammengehörigkeit kenntlich zu machen. Abbildung 4.5 zeigt, wie der Parallel Coordinate Plot drei Akteure mit sechs unterschiedlichen Parametern (A-F) darstellt.



**Abbildung 4.5:** Beispiel Parallel Coordinate Plot

Der Vorteil bei dieser Visualisierung liegt darin, dass ohne Probleme mehrere Dimensionen (Parameter) dargestellt werden können. Es lassen sich leicht sechs bis zehn Parameter gleichzeitig darzustellen. Die Komplexität der Parallel Coordinate Plots (Anzahl der Achsen) ist direkt proportional zu der Anzahl der Parameter [Spe07, S. 59]. Der Nachteil: Möchte man mehrere Akteure anzeigen, wird es sehr schnell unübersichtlich. Außerdem sind immer nur die Abhängigkeiten zwischen zwei nebeneinanderliegenden

Parametern vergleichbar und es lässt sich schlecht abschätzen, ob ein Akteur besser als der andere ist. Zusammenfassend ist zu sagen, dass der Parallel Coordinate Plot die Sichtbarkeit der Attribute (Parameter) unterstützt; d. h., es ist einfach, einen Überblick über die Zusammenhänge der unterschiedlichen Parameter der gegebenen Daten (Akteure) zu erhalten, ohne große kognitive Anstrengungen [Spe07, S. 59].

### Star Plot

Der *Star Plot*, im Deutschen auch als Netzdiagramm bekannt, gehört ebenso zu der Kategorie der Coordinate Plots. Er hat mit dem Parallel Coordinate Plot viele Gemeinsamkeiten. Auch hier wird ein Wert eines Parameters durch einen Punkt auf einer Achse abgebildet und die unterschiedlichen Parameterwerte eines Akteurs werden durch eine gerade Linie miteinander verbunden. Beim Star Plot werden jedoch die unterschiedlichen Achsen nicht parallel, sondern im Kreis angeordnet.

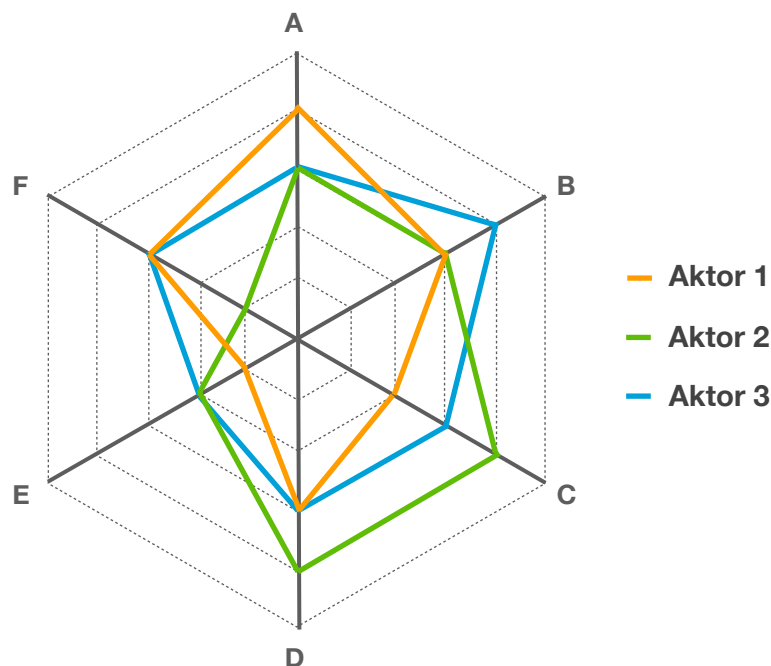


Abbildung 4.6: Beispiel Star Plot

Während der Parallel Coordinate Plot besser die Sichtbarkeit der einzelnen Attribute unterstützt, begünstigt der Star Plot eher die Sichtbarkeit eines Objektes; d. h., er eignet sich besser, um verschiedene Objekte (Aktoren) darzustellen und zu vergleichen. Auch die verschiedenen Parameter für einen Actor lassen sich gut erkennen. Der Nachteil bei diesem Diagramm liegt darin, dass nur wenig Aktoren gleichzeitig dargestellt werden können, da es sonst zu unübersichtlich wird. Abbildung 4.6 zeigt wie der Star Plot drei Aktoren und ihre sechs unterschiedlichen Parameter (A-F) darstellt.

### Scatter Plot

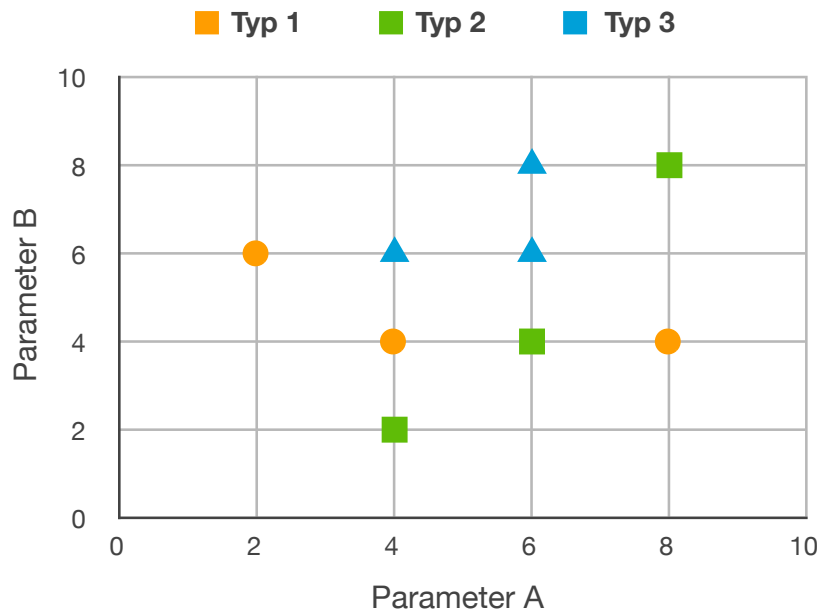
Der *Scatter Plot* (auch als Punkt- oder Streudiagramm bezeichnet) hat eine lange Tradition in der Informationsvisualisierung [Dem07]. Nach Schätzungen von *Edward Tufte*<sup>5</sup>, waren 1983 zwischen 70 und 80 Prozent aller Diagramme in wissenschaftlichen Arbeiten Scatter Plots [Tuf83]; d. h., der Scatter Plot ist den Nutzern, insbesondere Ingenieuren und Wissenschaftlern, sehr gut bekannt. Dies bildet eine wichtige Grundlage für die Akzeptanz der Visualisierung.

Mit ihm können beliebig viele Datenpunkte dargestellt werden. Dabei wird jeder Datenpunkt in einem kartesischen Koordinatensystem angeordnet. Es gibt zweidimensionale und dreidimensionale Scatter Plots. Erstere sind jedoch einfacher zu verstehen und deshalb auch die häufiger genutzten. Dreidimensionale Scatter Plots sollen hier aber nicht weiter behandelt werden, da sie die kognitiven Fähigkeiten des Nutzers zu sehr anstrengen und bei heutigen zweidimensionalen Ausgabegeräten keinen effektiven Nutzen bringen bzw. für kein besseres Verständnis sorgen.

Mit dem Scatter Plot ist es möglich, alle Treffer einer Suchanfrage gleichzeitig auf einen Blick darzustellen. Deshalb eignet er sich gut als Übersichtsdarstellung der gefunden Suchergebnisse. Darüber hinaus können Scatter Plots, neben den zwei Dimensionen für die Positionierung in dem kartesischen Koordinatensystem, noch weitere Dimensionen (Parameter) darstellen. Die Form eines Punktes und dessen Farbe können zwei zusätzliche Dimensionen kodieren. Abbildung 4.7 zeigt ein Beispiel des Scatter Plots

---

<sup>5</sup>US-amerikanischer Informationswissenschaftler und Grafikdesigner



**Abbildung 4.7:** Beispiel Scatter Plot

und seiner Kodierungsmöglichkeiten. Bei diesem Beispiel werden diesmal neun Aktoren mit ihren drei Parametern (A, B, Typ) dargestellt.

Ein Vorteil bei dieser Visualisierungsmethode liegt darin, dass für den Betrachter leicht die Relationen der Datenpunkte (Aktoren) im Datenbestand zu erkennen sind. Da die Scatter Plots, wie zuvor erwähnt, eine lange Vorgeschichte haben, in vielen wissenschaftlichen Arbeiten zum Einsatz kommen und leicht zu verstehen sind, stellen sie eine gute Methode zur Darstellung von hypervarianten Daten dar. Im Vergleich zu dem Parallel Coordinate Plot und dem Star Plot sind sie leichter zu verstehen und stellen besonders für den Ingenieur und Wissenschaftler eine bekannte Visualisierungsmethode dar. Ein weiterer Vorteil des Scatter Plots im Vergleich zu listenbasierten oder tabellarischen Visualisierungen, wie man sie aus aktuellen Suchsystemen kennt, liegt darin, dass sich Zusammenhänge und Trends sehr gut herausfinden lassen.

Ein Nachteil dieser Methode ist es, dass für die Darstellung der Abhängigkeiten zwischen mehreren einzelnen Parametern mehr als nur ein Scatter Plot notwendig ist. Während bei den Parallel Coordinate Plots und den Star Plots die Komplexität (An-

zahl der Parameter) direkt proportional zu der Anzahl der Achsen ist, steigt sie hier stärker an.

Bei der Visualisierung als Scatter Plot können außerdem zwei Probleme auftreten (siehe [Dem07, S. 14f]). Das erste Problem ist das der *Datapoint-Cluster*, bei dem es zu Überlappungen der Datenpunkte mit ähnlichen Ausprägungen bzgl. der visualisierten Dimension (Parameter) kommen kann. Mit steigender Anzahl angezeigter Datenpunkte (Aktoren) erhöht sich auch die Wahrscheinlichkeit des Auftretens von Datapoint Clustern. Das zweite Problem ist das der *Multiple-Datapoints*. Hierbei haben mehrere Datenpunkte die gleiche Ausprägungen bzgl. der angezeigten Dimensionen und kommen somit im Diagramm an der exakt gleichen Stelle vor. Damit ist nur der oberste, d. h. zuletzt gezeichnete Datenpunkt, sichtbar.

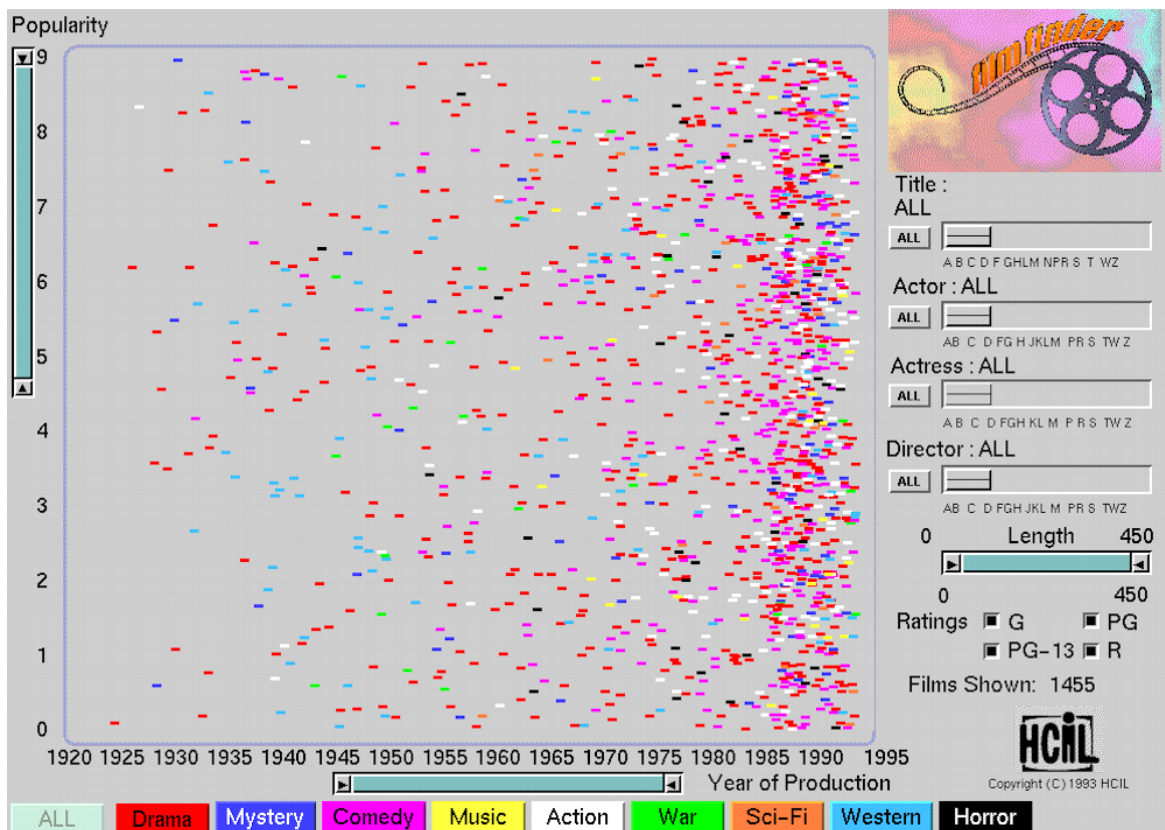


Abbildung 4.8: FilmFinder (vgl. [AS94])



Eine der bekanntesten interaktiven Beispielanwendungen des Scatter Plots ist der *FilmFinder* [AS94], mit dem der Inhalt einer Filmdatenbank dargestellt werden kann (siehe Abbildung 4.8). Jeder Film wird dabei durch verschiedenfarbige Punkte repräsentiert, wobei die Farbe des Punktes das Filmgenre kodiert. Die X-Y-Position auf dem Plot wird mittels der Metadaten Erscheinungsjahr und Popularität (Bewertung) bestimmt. Die restlichen Metadaten eines Filmes erscheinen über ein Pop-Up-Fenster, wenn der Nutzer auf den jeweiligen Film klickt. Der *FilmFinder* ist mit Hilfe des Konzepts der *Dynamic Queries* [Shn94] umgesetzt. Filtermöglichkeiten ermöglichen es, die Anzahl der Filme interaktiv, auf die für den Benutzer relevanten Einträge einschränken zu können.

### Bubble Chart

Eine Spezialform des Scatter Plots ist der *Bubble Chart* (Blasendiagramm). Er besitzt die gleichen Eigenschaften wie der Scatter Plot, verfügt aber mit der Größe der Blase über ein zusätzliches Merkmal, um eine weitere Dimension zu kodieren.

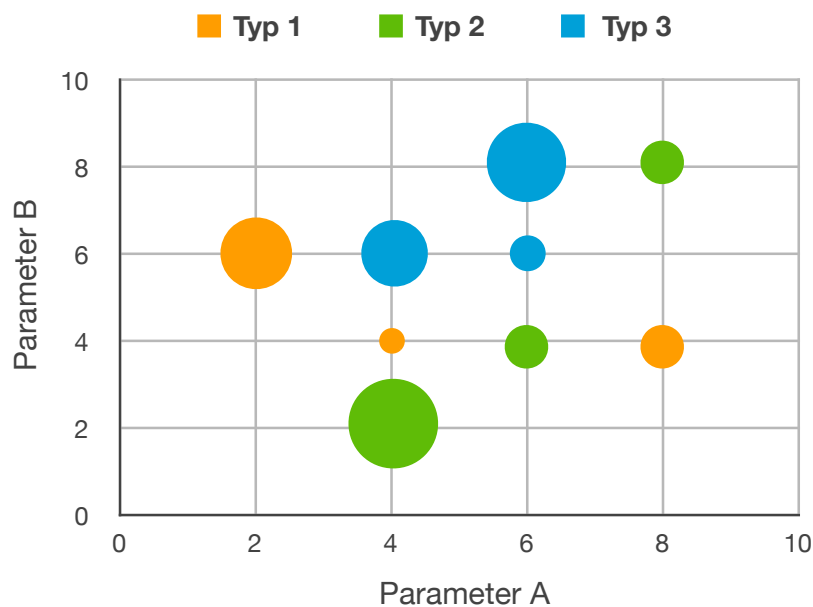


Abbildung 4.9: Beispiel Bubble Chart

Diese Form der Darstellung kann gewählt werden, wenn durch die Anzahl und den Abstand der Punkte kein übermäßiges Überlappen der Blasen auftritt (Datapoint-Cluster Problem [Dem07]). Besonders bei großen Datenmengen kann dies jedoch nicht immer garantiert werden. Abhilfe bei diesem Problem würde die Möglichkeit schaffen, den sich überlappenden Bereich vergrößern zu können. Gibt es eine zu große Ballung von Datenpunkten und damit Überlappungen der Blasen in dem Gebiet, kann dort bzgl. der X- und Y-Achsen rangezoomt werden. Dadurch lösen sich die Überlappungen und man erhält wieder eine übersichtliche Darstellung. Die Vor- und Nachteile des Bubble Chart sind ähnlich wie beim Scatter Plot.

### Bar Chart

Der *Bar Chart* (auch Balkendiagramm oder Säulendiagramm genannt) ist eine der gebräuchlichsten Diagrammtypen und kann ebenso für die Visualisierung von hyper-varianten Daten genutzt werden. Dabei kann ein Balken nur einen einzelnen Wert darstellen. Aber durch die Kombination mehrerer Balken ist es möglich, in einem Diagramm für ein Objekt mehrere Werte repräsentieren zu können. Der *Clustered Bar Chart* (Abbildung 4.10(a)) und der *Stacked Bar Chart* (Abbildung 4.10(b)) zeigen die Möglichkeit der Kombination mehrerer Balken in einem Diagramm.



Abbildung 4.10: Beispiel Clustered Bar Chart (a) und Stacked Bar Chart (b)

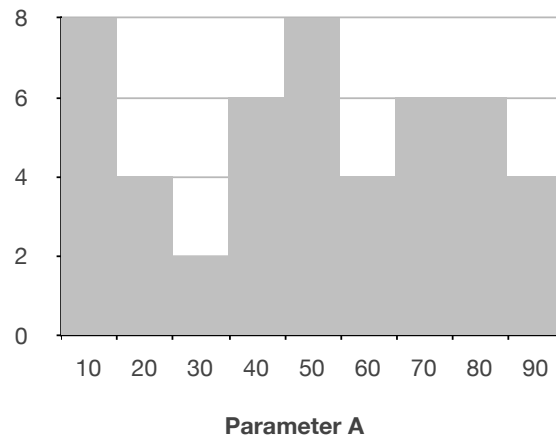
Bar Charts lassen sich nicht nur vertikal, sondern auch horizontal darstellen. Der Bar Chart gibt den Entwickler viele Möglichkeiten bei Wahl der Kodierung. Bei dem Stacked Bar Chart können die Größe der Säulen zum Beispiel als absolute oder relative Werte definiert sein. Die Gesamtgröße der Säule könnte beispielsweise für die Darstellung der Relevanz eines Aktors in Bezug auf die Suchanfrage (unter Berücksichtigung verschiedener Kriterien) genutzt werden. Der Vorteil bei dieser Visualisierungsmethode liegt darin, dass bei geringer Anzahl von zu vergleichenden Werten diese Form der Darstellung sehr übersichtlich ist und die verschiedenen Werte sich einfach miteinander vergleichen lassen. Möchte man aber eine größere Anzahl an Objekten (Aktoren) bzgl. eines Wertes miteinander vergleichen, ist der Bar Chart nicht besonders gut geeignet. Auch wenn sich mit Hilfe von aufeinander oder nebeneinander positionierten Balken mehrere Parameter gleichzeitig darstellen lassen, ist es jedoch unübersichtlich, da diese entlang einer einzigen Achse geordnet sind.

### **Linked Histograms / Attribute Explorer**

*Linked Histograms* (verlinkte Histogramme) sind eine weitere Technik, um hypervariante Daten zu repräsentieren. Jedes Histogramm stellt dabei eine Häufigkeitsverteilung über mögliche Werte von dem jeweiligen Attribut dar (siehe Abbildung 4.11); d. h., jede Höhe eines Balkens von einem Histogramm repräsentiert die Anzahl der Objekte, die diesen Wert besitzen bzw. in diesen Wertebereich fallen.

Die Anzahl der Objekte, die mit dieser Visualisierungsmethode dargestellt werden können, ist beliebig. Ebenso wie sie für die Darstellung von wenigen Objekten gut geeignet sind, können sie auch zur Repräsentation von vielen Objekten genutzt werden. Außerdem lassen sich sowohl qualitative als auch quantitative Daten darstellen.

Der *Attribute Explorer* (siehe Abbildung 4.12) ist dabei einer der bekanntesten Anwendungen dieser Technik und bietet durch zusätzliche Kodierungs- und Interaktionsmöglichkeiten besonderen Nutzen. Er erlaubt es, Randbedingungen bzw. Einschränkungen für jedes Histogramm und damit jedes Attribut anzugeben. Der Einfluss dieser Einschränkungen ist dann unmittelbar in den anderen Histogrammen sichtbar. Diese



**Abbildung 4.11:** Beispiel Histogramm

Interaktionstechnik wird auch als *Brushing* bezeichnet.

Im Gegensatz zum *FilmFinder*, bei dem nur Objekte dargestellt werden, die auch alle Kriterien erfüllen, werden hier immer alle Objekte auf einmal visualisiert. Wenn der Nutzer ein Kriterium ändert; d. h., ein Attribut einschränkt, sieht er durch die farbige Änderung der Objekte die Auswirkungen auf die Ergebnismenge in allen Histogrammen. Objekte, die alle Kriterien erfüllen, werden in allen Histogrammen mit einer bestimmten Farbe (im Beispiel siehe Abbildung 4.12 ist das weiß) dargestellt. Erfüllt ein Objekt ein oder mehrere Kriterien nicht, wird es mit einer anderen Farbe dargestellt. Somit kann der Nutzer erkennen, welche Objekte wie viele Kriterien erfüllen; d. h., es ist möglich zu erkennen, ob ein Objekt nur ein, zwei, drei etc. Kriterien verfehlt oder nicht. Handelt es sich bei der darzustellenden Datenmenge nur um eine geringe Anzahl an Objekten, ist es sogar möglich, einzelne Objekte im Histogramm identifizieren zu können (siehe Abbildung 4.12); d. h., ist ein Objekt in einem Histogramm ausgewählt, wird es auch in den anderen Histogrammen, welche andere Attribute darstellen, *ge-highlighted* (im Beispiel siehe Abbildung 4.12 gelb). Nachdem an den einzelnen parallel verlinkten Histogrammen die Einschränkungen bzgl. der jeweiligen Attribute getätigt wurden, ist es am Ende möglich, das Resultat der einzelnen Einschränkungen auf die gesamte Datenmenge zu sehen.

Der Attribute Explorer begünstigt, ebenso wie der Parallel Coordinate Plot, die Sicht-

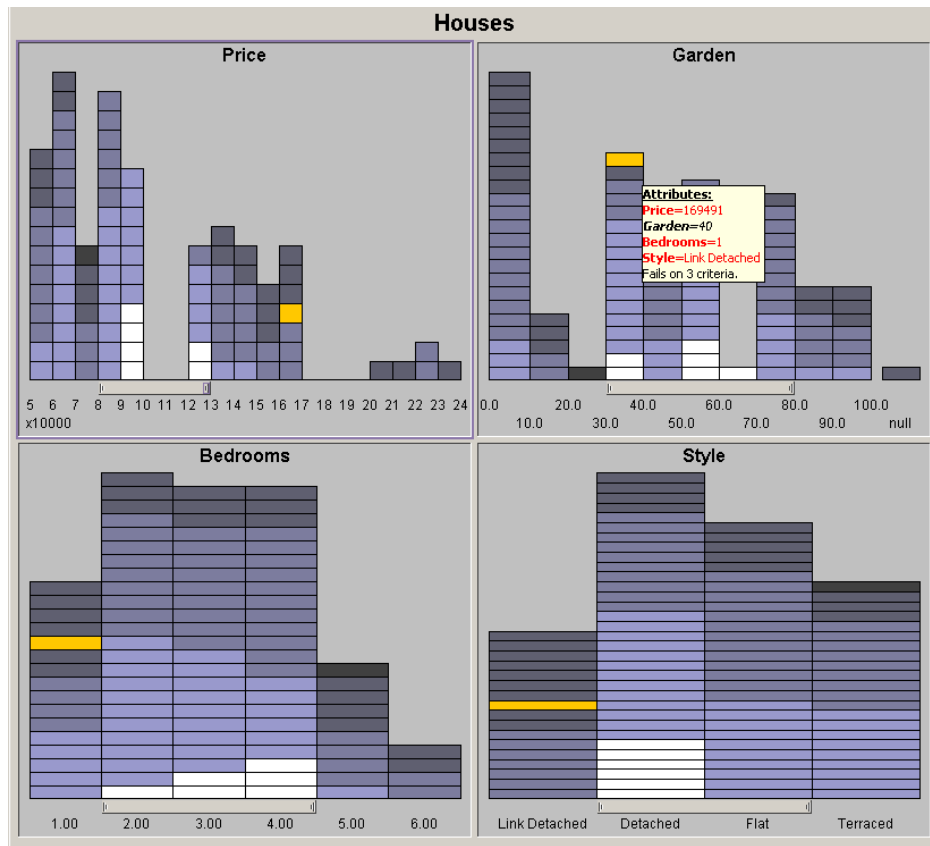


Abbildung 4.12: Attribute Explorer

barkeit der einzelnen Attribute; d. h., er macht die interaktive Exploration der Daten möglich und lässt somit die Beziehungen zwischen den Attributen der hypervarianten Daten erkennen. Auch die Häufigkeitsverteilung der Objekte nach den Attributen ist zu sehen. Das ursprüngliche Ziel des Attribute Explorers war es, ein Tool bereitzustellen, um die Auswahl von einem Objekt aus vielen, basierend auf deren Attributen, zu ermöglichen. Jedoch hat sich gezeigt, dass sich der Attribute Explorer auch für die Exploration (Browsen) von großen Datenmengen gut eignet (siehe [Spe07, S. 62]).

### Table Lens

*Table Lens* (siehe [Spe07, S. 105f]) ist eine Visualisierung in Tabellenform mit integrierter Balkendarstellung. Diese Visualisierungsmethode ermöglicht die Exploration und

Interaktion mit großen Datensätzen. Mit Hilfe der Balkendarstellung ist es Nutzern schnell möglich, eine Übersicht von großen Datenbeständen zu erhalten. Außerdem können damit Muster, Ausreißer und Beziehungen einfacher erkannt und gleichzeitig Details zu ausgewählten Objekten betrachten werden (siehe Abbildung 4.13).



Abbildung 4.13: Table Lens

Die Table Lens Darstellung eignet sich besonders für numerische und kategoriale Daten. Dabei repräsentieren die Zeilen die verschiedenen Objekte (z. B. Aktoren) und die Spalten deren Attribute (Parameter). Mit Hilfe der *Fisheye* Interaktionstechnik ist es möglich, ein oder mehrere Zeilen detaillierter zu betrachten; d. h., zusätzlich zu der Balkendarstellung auch textliche Information über die exakten Werte dieser Zeilen zu erhalten. Dadurch, dass der Table Lens in der Übersichtsdarstellung nur die Balken anzeigt, ist es möglich, eine große Datenmenge auf geringsten Platz darstellen zu können. Dies ist nicht nur ein Skalierungsvorteil, sondern bedeutet auch bei der Exploration großer Datenmengen einen Vorteil, da die große Menge der schmalen Balken vom Benutzer wesentlich schneller überflogen werden können, als eine vergleichbare Menge an textlich repräsentierten Zahlen. Der Table Lens bietet mit dem Sortieren einer Spalte anhand ihrer Werte eine zusätzliche Interaktionsmöglichkeit an, welche die Exploration der hypervarianten Daten des Weiteren erleichtern.

#### 4.3.2.2 Bewertung

Die umgesetzte Visualisierung muss die gefundenen Ergebnisse der Aktorsuche grafisch übersichtlich darstellen können; d. h., sie muss in der Lage sein, sowohl mit vielen als auch mit wenigen Ergebnissen (Aktoren) umgehen zu können. Die Visualisierung muss außerdem mehrere Parameter gleichzeitig überschaubar abbilden können, ohne dabei zu aufdringlich oder zu komplex zu wirken. Auch die Abhängigkeit der einzelnen Kriterien muss zum Ausdruck kommen. Zum Beispiel, wenn man den Parameter A ändert, wie wirkt sich das auf den Parameter B und schließlich auf die Ergebnisse aus. Da die Visualisierung Übersichtscharakter haben soll, muss sie Ausreißer, Cluster und Muster gut darstellen können. Aber auch der Vergleich mehrerer Objekte untereinander muss möglich sein.

Der Parallel Coordinate Plot eignet sich einerseits gut, um viele Objekte mit mehreren unterschiedlichen Parametern darzustellen; andererseits ist er aber nicht besonders intuitiv verständlich. Die Sichtbarkeit der Attribute ist hier gut zu erkennen, aber er kann keine gute Objekt-Sichtbarkeit gewährleisten. Es ist extrem schwierig, die einzelnen Attribute eines Objektes zu erkennen und wie die Objekte in Beziehung zueinander stehen.

Der Star Plot dagegen kann leider nicht viele Aktoren auf einmal darstellen und eignet sich somit nicht für eine Übersichtsdarstellung. Da er sich aber hinsichtlich des Vergleichs weniger Objekte bzgl. dessen Parameter gut eignet, kann er evtl. zum näheren Vergleich dieser genutzt werden.

Der Scatter Plot und der Bubble Chart bieten hinsichtlich der Übersichts- und Vergleichsfunktion der verschiedenen Aktoren den größten Vorteil. Sie eignen sich gut, eine große Menge von Aktoren gleichzeitig darzustellen, ohne unübersichtlich zu wirken und bieten trotzdem eine gute Objekt- und Attribut-Sichtbarkeit. Da der Scatter Plot, wie zuvor erwähnt, eine lange Vorgeschichte hat und in vielen wissenschaftlichen Arbeiten zum Einsatz kam, ist er leicht und intuitiv zu verstehen. Aufgrund der ähnlichen Eigenschaften ist der Bubble Chart ebenfalls sehr simpel zu interpretieren. Der Bubble Chart hat zugleich den Vorteil, eine zusätzliche Dimension darstellen zu können. Beide Methoden sind somit besonders gut für die Visualisierung der Sucher-

gebnisse geeignet. Sie sind außerdem die beiden einzigen Visualisierungsmöglichkeiten die für die Darstellung einer beliebig große Anzahl von Objekten geeignet sind.

Auch der Bar Chart in seinen beiden vorgestellten Varianten ist nicht die optimale Lösung für die Ergebnisdarstellung der Aktorsuche. Es ist mit ihm nicht möglich, eine große Menge von Daten übersichtlich darzustellen. Lediglich als alternative Visualisation, wenn die Ergebnismenge schon weiter eingeschränkt ist, kann er genutzt werden.

Der Attribute Explorers ist durch seine Interaktionsmöglichkeiten gut zur Exploration und Analyse von mittleren bis großen Datenmengen geeignet. Die Verteilung der einzelnen Werte je Attribut und die Abhängigkeiten der Objekte bzgl. ihrer Attribute sind sehr gut erkennbar (Attribut-Sichtbarkeit). Der Attribute Explorer ermöglicht es, durch seine direkte Manipulation Suchanfragen zu stellen und die Ergebnisse unmittelbar sichtbar zu machen. Es ist jedoch mit dieser Visualisierung sehr schwierig, die einzelnen Attribute eines Objektes zu erkennen und wie die Objekte zueinander in Beziehung stehen (Objekt-Sichtbarkeit). Seine verlinkten Histogramme mit ihren Häufigkeiten und Verteilungen sind sehr abstrakt und im Gegensatz zum Scatter Plot bzw. Bubble Chart wenig intuitiv und würden eine gewisse Zeit der Einarbeitung benötigen. Außerdem ist er auf Grund der Objekt-Abhängigkeiten zwischen den verlinkten Histogrammen und seinen Interaktionsmöglichkeiten schwierig umzusetzen.

Der Table Lens eignet sich auf Grund seiner Balkendarstellung gut für die Exploration und Interaktion mit großen Datensätzen, mit dessen Hilfe es dem Nutzer schnell möglich ist, eine Übersicht der Datenbestände zu erhalten. Muster, Ausreißer und Beziehungen können somit leicht erkannt werden und gleichzeitig ist es möglich, Details zu ausgewählten Objekten anzeigen zu lassen.

Da der Scatter Plot die meisten Vorteile bietet und auch bei großen Datenmengen übersichtlich und leicht zu verstehen ist, wurde er für die Umsetzung in der Webanwendung zur Aktorsuche ausgewählt. Die Nachteile, die er aufweist, werden durch zusätzliche Interaktionsmöglichkeiten (siehe folgenden Abschnitt) beseitigt.



### 4.3.2.3 Ergebnis

Für die Visualisierung der Ergebnisse der Aktorsuche wurde sich für den Bubble Chart entschieden. Dieser soll primär eine Übersichtsfunktion bieten, um den Nutzer mit wertvollen Informationen (z. B. Ausreißer, Cluster etc.) bzgl. der gezielten Suche zu versorgen und bei der freien Exploration (Browsen) einen Überblick über die gesamten Daten zu bieten.

Die Visualisierung, die umgesetzt werden soll, ist im Gesamtkonzept stark an den *FilmFinder* (siehe [AS94]) angelehnt. Das Suchinterface (siehe Kapitel 4.3.1) auf der linken Seite soll dabei die Filtermöglichkeit des *FilmFinders* gewährleisten, während die Visualisierung die gefundenen Ergebnisse darstellt. Abbildung 4.14 verdeutlicht die entworfene Visualisation.

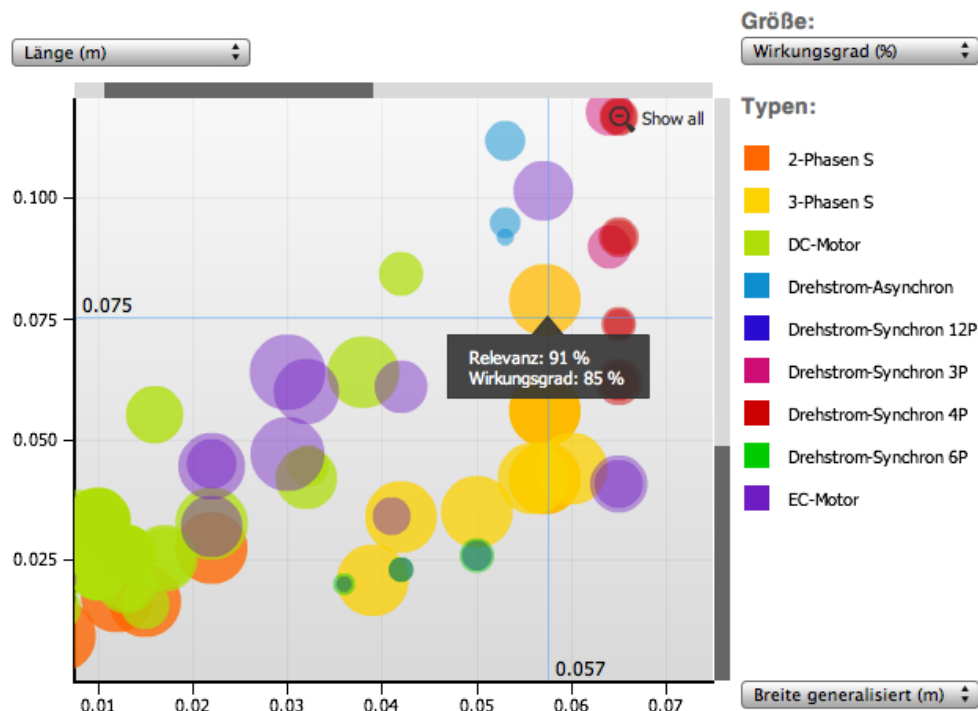


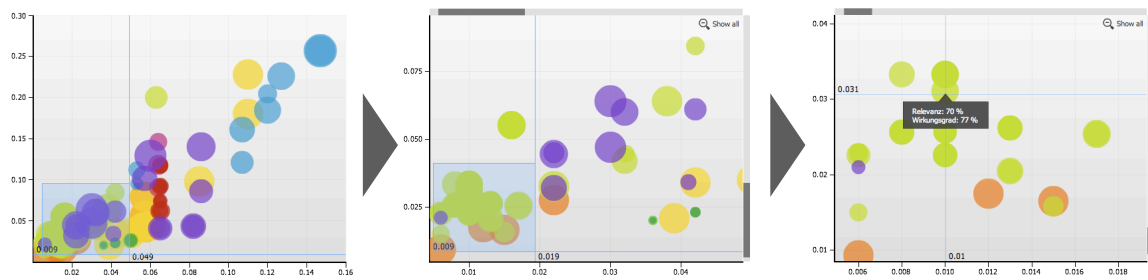
Abbildung 4.14: Bubble Chart als umgesetzte Visualisierung

Ein Nachteil des *FilmFinders* im Vergleich zu dem Attribute Explorer ist, dass er nur die Objekte darstellt welche alle, durch den Filter angegebenen Kriterien erfüllen. In

der hier realisierten Webanwendung werden aufgrund des verwendeten Suchalgorithmus (Skyline-Algorithmus) auch Ergebnisse dargestellt, welche ein Suchkriterium nur knapp verfehlen.

Der Bubble Chart ermöglicht es, gleichzeitig fünf verschiedene Dimensionen anzuzeigen: Die X- und Y-Koordinaten, die Größe, Farbe und Form des Punktes. Während die X- und Y-Koordinaten sowie die Größe quantitative Werte wiedergeben können, ist die Kodierung durch die Farbe oder Form besser für qualitative Werte geeignet. Der umgesetzte Bubble Chart ermöglicht es aktuell vier Dimensionen anzuzeigen, wobei er leicht um die fünfte Dimension (Form) erweitert werden kann (siehe dazu später mehr). Um ein Vergleichen aller vorhandenen Parameter zu realisieren, musste die Möglichkeit geboten werden, die angezeigten Dimensionen ändern zu können. Mit Hilfe von Auswahlboxen an den jeweiligen Achsen lassen sich alle quantitativen Parameter der Aktoren auswählen und somit in der Visualisierung grafisch darstellen. Die Auswahlbox für die Größe der Blasen bietet zudem die Möglichkeit, die Relevanz der Ergebnisse oder gar keinen Wert auszuwählen. Bei Letzterem würden die Blasen in dem Diagramm alle gleich groß dargestellt. Die Entscheidung, welche Parameter angezeigt werden sollen, liegt ganz allein beim Benutzer. In der umgesetzten Visualisierung repräsentiert die Farbe die verschiedenen Aktortypen. Jedem Aktortyp ist dabei eine bestimmte Farbe zugeordnet, um die Wiedererkennung später zu erleichtern. Auf die Kodierungsmöglichkeit durch unterschiedliche Formen wurde vorerst verzichtet, damit der zu entwickelnde Prototyp nicht zu komplex und überladen wird. Es wäre aber durchaus denkbar, eine zusätzliche Dimension (für qualitative Daten) damit zu repräsentieren. Wie in dem vorhergehenden Abschnitt beschrieben, kann es bei Scatter Plots und Bubble Charts zu den Problemen Datapoint-Cluster und Multiple-Datapoints kommen. Um diese Schwierigkeiten der Überlappungen bzw. Überdeckungen zu beseitigen, wird die Möglichkeit des Ranzoomens angeboten. Der Nutzer kann durch Klicken und Ziehen mit der Maus einen Bereich, der vergrößert werden soll, auswählen (siehe Abbildung 4.15).

Interessiert sich der Nutzer für die exakten Werte des dargestellten Aktors, kann er mit



**Abbildung 4.15:** Zoom- und Scroll-Funktion in der Visualisierung

der Maus über den gewünschten Aktor gehen und erhält unmittelbar einen Tooltip<sup>6</sup> über den exakten Wert und die Relevanz des Aktors (siehe Abbildung 4.14).

Während einiger Tests und heuristischer Evaluationen stellten sich bei bestimmten Designentscheidungen Probleme heraus. Diese Problempunkte wurden beim Redesign jedoch behoben. Zum Beispiel erwies sich die Position der Auswahlboxen für die verschiedenen Dimensionen anfangs als nicht optimal. Die Auswahlboxen für die X- und Y-Achse und die Größe der Blasen waren zu Beginn oberhalb der Visualisierung angebracht. Sie wurden damit sehr prominent angezeigt, konnten aber keinen Zusammenhang zwischen dem Auswählen der Dimension und dem Anzeigen herstellen; d. h. die Benutzung war für den Nutzer nicht sehr intuitiv, da nicht leicht zu erkennen war, welche Auswahlbox z. B. für die X-Achse genutzt werden musste. Ein zweites Beispiel war die Idee, die Transparenz der Blasen für die Relevanz der Aktoren zu verwenden. Je relevanter ein Aktor für die gestellte Suchanfrage ist, desto sichtbarer sollte er sein. Mittels heuristischer Evaluation zeigte sich dabei folgende Problematik. Aufgrund des nicht auszuschließenden Problems der Multiple-Datapoints kann es passieren, dass mehrere Aktoren übereinander dargestellt werden und sich somit die Transparenzen der einzelnen Aktoren überlagern. Das Resultat war, dass die Relevanzen der einzelnen Aktoren falsch dargestellt und somit die Ergebnisse verfälscht wurden. Diese Aufschlüsse haben dazu geführt, dass die Transparenz der Blasen einheitlich festgelegt ist, ohne durch einen anderen Parameter beeinflussbar zu sein.

<sup>6</sup>ist ein automatischer Hinweis (Kurzinfo) in einem kleinen schwebenden Fenster

### 4.3.3 Ergebnistabelle

Die Ergebnistabelle soll zusätzlich zu der Visualisierung für den Überblick eine detaillierte Darstellung der vorhandenen Ergebnisse mit allen Parametern liefern und das Vergleichen, Sortieren und gegebenenfalls weitere Aktionen ermöglichen.

Die Tabellendarstellung wurde gewählt, da sie eine bekannte und weit verbreitete Form der Darstellung hypervarianter Daten ist. Sie ermöglicht es, viele Werte und ihre Parameter gleichzeitig übersichtlich darzustellen und bietet zudem gute Vergleichsmöglichkeiten. Zusätzliche Interaktionsmöglichkeiten können durch die Sortierung der Daten nach bestimmten Parametern angeboten werden. Die Darstellung der Ergebnisse durch eine Tabelle wurde von mehreren Nutzern in der Anforderungsanalyse erwünscht. Dies begründet sich mit ihrem weiten Verbreitungsgrad und ihrer Beliebtheit. Um eine bessere Übersicht sowie ein vorteilhafteres Vergleichen und Erfassen der Tabelle zu gewährleisten, wurde eine Balkendarstellung, ähnlich wie beim Table Lens, integriert (siehe Abbildung 4.16).

Im Unterschied zum Table Lens werden die Details nicht erst nach dem Auswählen angezeigt. Die umzusetzende Tabellendarstellung muss als Detailansicht dienen und demzufolge von Anfang an Einzelheiten anzeigen. Für ein besseres Verständnis des Nutzens dieser integrierten Balkendarstellung wurde zum Vergleich eine normale Tabelle mit üblicher Zahlendarstellung illustriert (siehe Abbildung 4.17). Betrachtet man diese beiden Abbildungen, so wird schnell der Vorteil dieser erweiterten Tabelle mit Balkendarstellung erkennbar.

Bei der integrierten Balkendarstellung sind zwei Fälle zu unterscheiden. Zum einen gibt es Balken für die Kodierung von absoluten Zahlenwerten und zum anderen für die Bereichskodierung plus Zahlenwertkodierung. Wird nach einem bestimmte Bereich für ein Parameter gesucht (größer, kleiner oder im Bereich) ist es für den Nutzer wichtig zu erkennen, welche Akteure in einem Bereich liegen, welche den Bereich knapp oder stark verfehlen und wie der eigentliche Wert des betrachteten Parameters für die Akteure sind. Abbildung 4.16 illustriert für den Parameter “Breite generalisiert” eine solche Balkendarstellung für die Bereichskodierung. Der grüne Abschnitt des Balkens gibt

1 bis 25 von 50 Einträgen

Anfang12>Ende

Filtern

Relevanz	Typ	Hersteller	Drehzahl (1/s)	Wirkungsgrad (%)	Leistung (W)	Breite generalisiert (m)▲	Länge (m)	Masse (kg)
99	EC-Motor	ebm-papst	75	45	2.7	<div><div></div></div>	0.0341	0.12
99	DC-Motor	Crouzet	33	52	15	<div><div></div></div>	0.0844	0.4
98	Drehstrom-Synchron 12P	Berger Lahr	4	29	0.8	<div><div></div></div>	0.023	0.15
98	Drehstrom-Synchron 6P	Berger Lahr	8	30	1.8	<div><div></div></div>	0.023	0.15
98	3-Phasen S	Color-Tech	31	85	2.34	<div><div></div></div>	0.034	0.2
98	EC-Motor	Nanotec	67	62	52.5	<div><div></div></div>	0.061	0.45
97	Drehstrom-Synchron 12P	Berger Lahr	4	35	1.3	<div><div></div></div>	0.026	0.2
96	3-Phasen S	Color-Tech	47	85	3.54	<div><div></div></div>	0.035	0.23
93	Drehstrom-Synchron 6P	Berger Lahr	8	41	2	<div><div></div></div>	0.026	0.2
92	Drehstrom-Asynchron	Gefegneckar	43	47	12	<div><div></div></div>	0.112	1
92	Drehstrom-Asynchron	Gefegneckar	20	20	3	<div><div></div></div>	0.092	0.7
91	Drehstrom-Asynchron	Gefegneckar	45	36	10	<div><div></div></div>	0.095	0.7
90	3-Phasen S	Color-Tech	28	85	4.44	<div><div></div></div>	0.042	0.4
89	EC-Motor	Nanotec	47	71	100	<div><div></div></div>	0.1016	1.5
88	2-Phasen S	Berger Lahr	16	85	29.8	<div><div></div></div>	0.042	0.5
86	2-Phasen S	Berger Lahr	13	85	29.8	<div><div></div></div>	0.056	0.7
84	2-Phasen S	Berger Lahr	11	85	89.3	<div><div></div></div>	0.079	1.05
82	2-Phasen S	Berger Lahr	15	85	62.5	<div><div></div></div>	0.056	0.7
81	2-Phasen S	Berger Lahr	17	85	62.5	<div><div></div></div>	0.042	0.5
81	2-Phasen S	Berger Lahr	15	85	89.3	<div><div></div></div>	0.056	0.7
78	3-Phasen S	Berger Lahr	43	85	209.9	<div><div></div></div>	0.079	2

Abbildung 4.16: Erweiterte Tabelle mit Balkendarstellung

Relevanz	Typ	Hersteller	Drehzahl	Wirkungsgrad	Leistung	Breite generalisiert	Länge	Masse	Volumen	Dichte
100%	DC-Motor	Maxon	118	66	0.75	0.013	0.0205	0.015	0.00272102	5.51264
100%	DC-Motor	Maxon	180	74	1.2	0.013	0.0205	0.015	0.00272102	5.51264
100%	DC-Motor	Maxon	129	79	2.5	0.017	0.0254	0.027	0.00576531	4.68318
100%	DC-Motor	Maxon	168	72	4.5	0.017	0.0254	0.026	0.00576531	4.50973
100%	DC-Motor	Faulhaber	325	50	0.11	0.006	0.015	0.002	0.000424116	4.71569
100%	DC-Motor	Faulhaber	213	58	0.54	0.015	0.0158	0.013	0.0027921	4.65599
100%	DC-Motor	Faulhaber	133	86	3.85	0.022	0.0326	0.061	0.0123923	4.92241
100%	DC-Motor	Faulhaber	90	73	27.3	0.032	0.042	0.175	0.0337785	5.18081
100%	DC-Motor	Crouzet	62	48	3	0.0322	0.0446	0.096	0.036319	2.64324
100%	DC-Motor	Crouzet	33	52	15	0.042	0.0844	0.4	0.117486	3.40466
100%	DC-Motor	Crouzet	25	63	27	0.063	0.2	0.84	0.623451	1.34734
100%	DC-Motor	Faulhaber	112	85	226	0.038	0.064	0.4	0.0725835	5.51089
100%	EC-Motor	Maxon	602	39	1.2	0.006	0.021	0.0028	0.00059376	4.71571
100%	EC-Motor	Maxon	467	73	12	0.022	0.032	0.067	0.0121643	5.50792
100%	EC-Motor	Maxon	160	79	20	0.022	0.0445	0.085	0.0169159	5.02486
100%	EC-Motor	Maxon	188	77	80	0.032	0.06	0.27	0.048255	5.59527
100%	EC-Motor	Maxon	297	87	100	0.03	0.047	0.165	0.0332224	4.96653
100%	EC-Motor	Maxon	275	89	200	0.03	0.064	0.271	0.045239	5.99041
100%	EC-Motor	Maxon	52	89	400	0.06	0.1292	2.45	0.365305	6.70672
100%	EC-Motor	Nanotec	80	59	3.8	0.022	0.045	0.07	0.017106	4.09213

Abbildung 4.17: Normale Tabelle mit Zahlendarstellung

den gesuchten Bereich an, während der weiße Abschnitt den nicht gesuchten Bereich darstellt. Der schwarze Strich repräsentiert den tatsächlichen Wert des jeweiligen Parameters. Bei dieser Balkendarstellung liegt der Fokus mehr auf der Bereichskodierung. Möchte der Nutzer den exakten Wert des Parameters angezeigt bekommen, muss er mit der Maus über den Balken für den Parameter des Aktors gehen, und er erhält unmittelbar einen Tooltip mit der gewünschten Information.

Die Ergebnistabelle bietet neben den zwei verschiedenen Balkendarstellungen außerdem die Möglichkeit, nach mehreren Spalten (Parametern) zu sortieren. Durch Drücken der Shift-Taste und Klicken auf eine Spalte können somit auch mehrere Spalten ausgewählt werden, nach denen sortiert werden soll. Durch Klicken auf eine Spalte wird nur diese eine Spalte für die Sortierung genutzt.

Die entworfene Tabellendarstellung unterstützt auch das Filtern der enthaltenen Ergebnisse. Primär soll natürlich das Suchinterface für die Suche bzw. das Filtern verwendet werden. Möchte der Nutzer temporär sehen, welche Aktoren es zum Beispiel vom Hersteller *Nanotec* gibt oder welche vom Typ *EC-Motor* sind, kann er nach diesem filtern. Auch das Filtern nach Zahlenwerten ist möglich. Diese Filtermöglichkeit wirkt sich aber nur auf die Tabelle, nicht auf die Visualisierung aus.

Neben der Balkenkodierung für den Suchbereich und dem absoluten Wert gibt es noch eine Farbkodierung für die einzelnen Aktortypen. Die Farben, welche in der Visualisierung für die verschiedenen Typen genutzt wurden, werden auch in der Tabellendarstellung verwendet. Dies ermöglicht zum einen das schnellere Erfassen der unterschiedlichen Typen in der Detailansicht (Tabelle) und zum anderen wird damit der Zusammenhang zwischen den Aktoren in der Visualisierung und der Tabelle besser deutlich.

Wie in der Anforderungsanalyse beschrieben, wurde von den Nutzern die Möglichkeit gewünscht, bei der Ergebnisdarstellung Zugriff auf Links zu den Kontaktdaten des Herstellers sowie zu Datenblättern zu haben. Da die Aktor-Datenbank bisher keine Links zu Bestellinformationen und Datenblättern des jeweiligen Aktors aufweist, erwies sich eine *Google*-Suche als am sinnvollsten. Klickt man in der Ergebnistabelle auf den Hersteller, wird unmittelbar eine Suche nach dem Hersteller und dem Aktortyp gestartet.

Neben den in der Tabelle verwendeten Aktor-Farben wurde bei der Farbwahl darauf geachtet, dass sie nicht zu aufdringlich erscheint. Die abwechselnd gefärbten Zeilen helfen dem Nutzer beim Scannen der Ergebnistabelle und dem Unterscheiden der einzelnen Zeilen (Aktoren). Auch jene Zeile, die sich unter dem Mauscursor befindet und die Spalten, nach denen sortiert wird, werden zusätzlich farblich hervorgehoben.

## 4.4 Designergebnis

Das Gesamtergebnis des Screen-Designs und des detaillierten UI-Designs ist in Abbildung 4.18 dargestellt. Bei dieser Abbildung handelt es sich um einen Screenshot<sup>7</sup> des aktuellen (11.05.2009) Designs der Webanwendung.

---

<sup>7</sup>graphische Ausgabe des dargestellten Bildschirminhalts

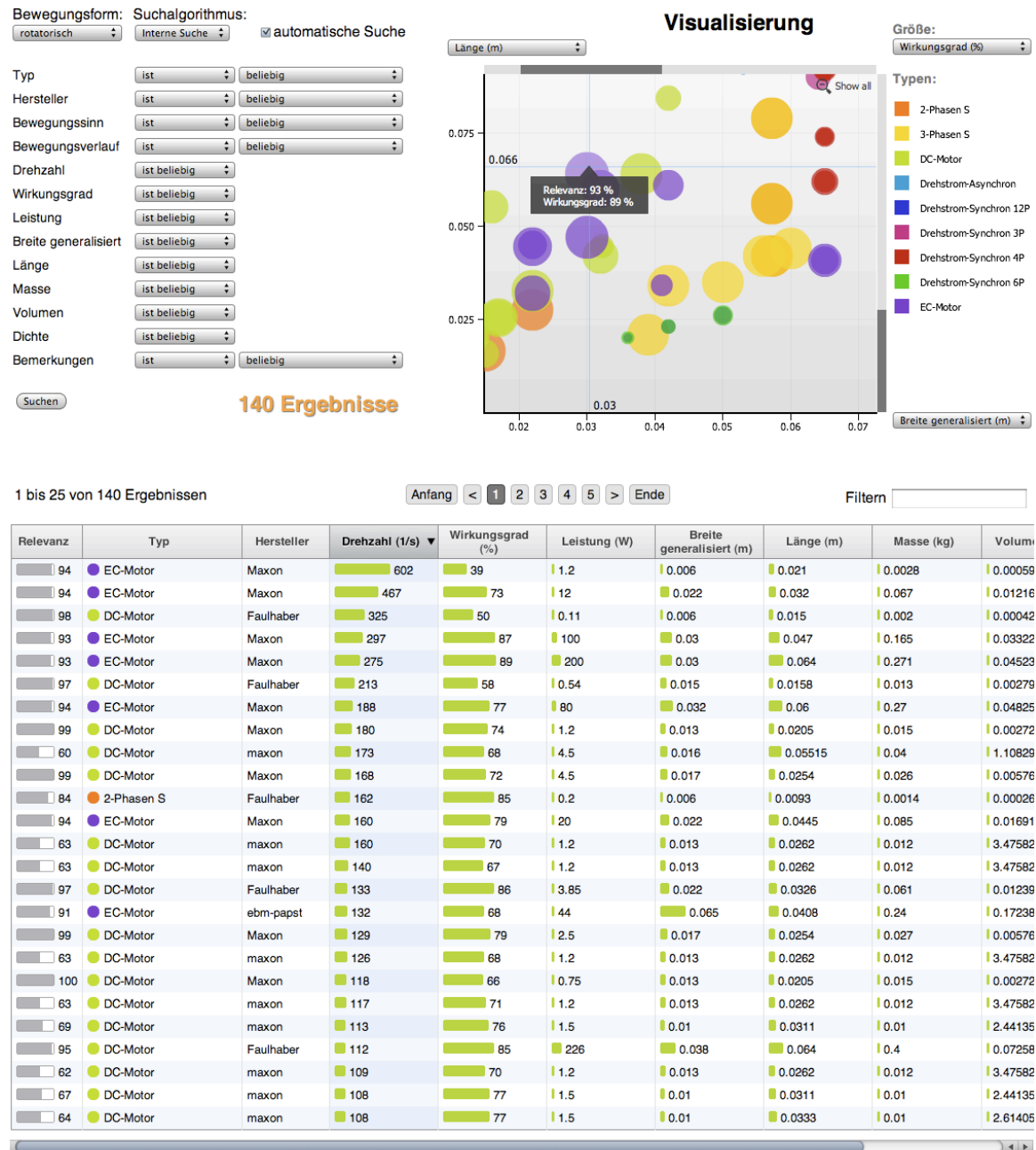


Abbildung 4.18: Designergebnis



# Kapitel 5

## Entwicklung/Umsetzung

Bei der Umsetzung wurde darauf geachtet, dass die zu erstellende Webanwendung bezüglich Interaktivität und Design einer Desktopanwendung ebenbürtig ist. Für der Umsetzung der im Kapitel 4 gestalteten Webanwendung kamen folgende Web-Technologien zum Einsatz.

### PHP

*PHP: Hypertext Preprocessor (PHP)* (siehe [Möh06]) ist eine an C<sup>1</sup> angelehnte Open-Source Skriptsprache, die vor allem zur Erstellung von dynamischen Webseiten oder Webanwendungen verwendet wird. Der PHP-Code wird dabei serverseitig verarbeitet. Der Ablauf beim Besuchen einer PHP-Seite sieht wie folgt aus (siehe Abbildung 5.1).

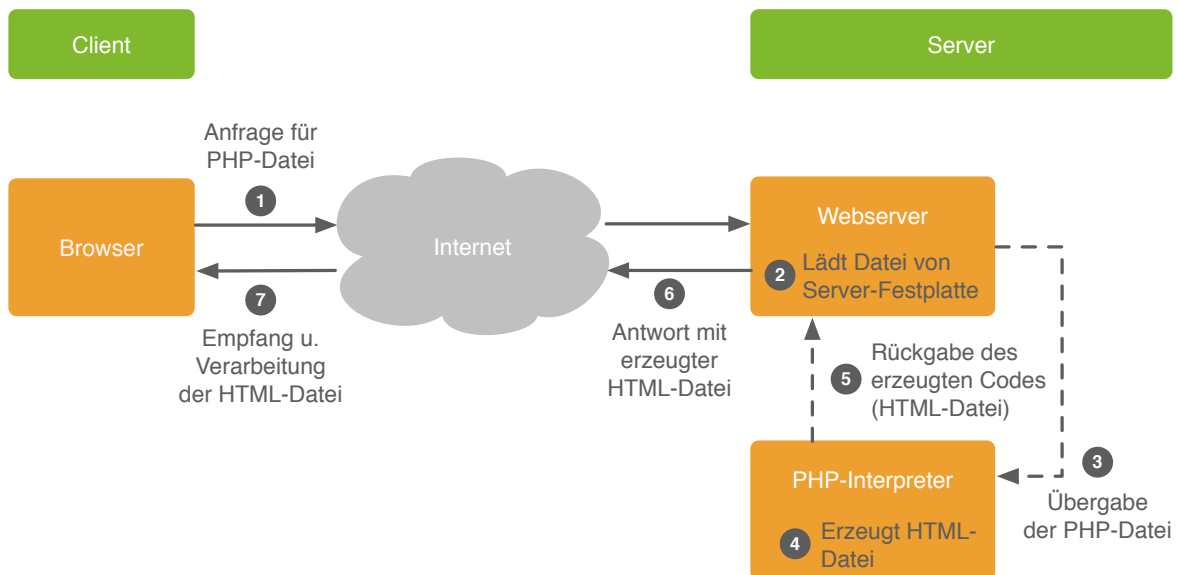
1. Der Browser (Client) fordert eine PHP-Datei an. Dies erfolgt über die Eingabe einer URL<sup>2</sup> bzw. durch Anklicken eines Links. Die Anfrage wird über das Internet an den entsprechenden Server weitergeleitet.
2. Der Webserver (Server) nimmt die Anfrage entgegen und lädt die PHP-Datei von seiner Festplatte.
3. Dann übergibt er die PHP-Datei an den PHP-Interpreter.

---

<sup>1</sup>imperative Programmiersprache

<sup>2</sup>identifiziert eine Internet Ressource (Synonym für Internetadresse)

4. Der PHP-Interpreter arbeitet das PHP-Skript ab und erzeugt eine HTML-Datei.
5. Anschließend gibt der Interpreter das Ergebnis seiner Arbeit (die erzeugte HTML-Datei) an den Webserver zurück.
6. Der Webserver liefert danach die erzeugte HTML-Datei an den Browser (Client) zurück.
7. Der Browser empfängt die HTML-Datei und verarbeitet sie. Dabei werden unter Umständen externe JavaScripts, Bilder und *CSS*-Dateien nachgeladen.



**Abbildung 5.1:** Ablauf beim Besuchen einer PHP Seite

Die Implementierung erfolgte in einer *Mac Apache MySQL PHP (MAMP)*-Umgebung unter dem Einsatz der PHP-Version 5.2.6.

## XHTML

*Extensible Hypertext Markup Language (XHTML)* ist eine textbasierte Auszeichnungssprache zur Strukturierung und semantischen Auszeichnung von Inhalten. XHTML

war eine von dem *World Wide Web Consortium (W3C)*<sup>3</sup> im Jahr 2000 komplette Neuformulierung der *Hypertext Markup Language (HTML)* 4.01 gültigen Befehle in *Extensible Markup Language (XML)*. Sie ist hinsichtlich des Syntax wesentlich strikter als HTML und erlaubt beispielsweise nur noch kleingeschriebene und geschlossene Tags<sup>4</sup>. Dies ermöglicht eine konsistent interpretierbare Dokumentstruktur und kann somit maschinell eingelesen und analysiert werden.

## CSS

Da sich XHTML nur für das Strukturieren der Inhalte, nicht aber das Steuern des Aussehens der einzelnen Elemente eignet, ist es wichtig, von *Cascading Style Sheets (CSS)* Gebrauch zu machen. CSS ermöglicht die konsequente Trennung des Aussehens von der Struktur bzw. dem Inhalt; d. h., alle Anweisungen, die das Design und die Formatierung der Webanwendung betreffen, befinden sich nicht direkt im HTML-Code. Dabei gibt es drei Möglichkeiten CSS in HTML-Dokumente zu integrieren.

- Die CSS-Anweisungen direkt als Attribut in einem HTML-Tag speichern.
- Die CSS-Anweisungen in den Kopfbereich eines HTML-Dokuments einbinden.
- In den Kopfbereich des HTML-Dokuments auf eine externe CSS-Datei verweisen.

Die letzte Möglichkeit setzt die Trennung von Design und Struktur bzw. Inhalt am konsequentesten durch und ist die Methode, die fast ausnahmslos in der entwickelten Webanwendung umgesetzt wurde. Ein weiterer Vorteil dieser Methode ist es, dass mehrere HTML-Dokumente auf diese CSS-Dateien zugreifen können; d. h., Änderungen am Stylesheet wirken sich somit global aus.

## Javascript

Während XHTML und CSS für die Struktur und Darstellung von Webseiten verwendet wird, kann JavaScript für die Umsetzung der interaktiven Elemente einer Webseite

---

<sup>3</sup>Gremium zur Standardisierung der im Internet verwendeten Techniken

<sup>4</sup>Tags dienen dazu, Textelemente auszuzeichnen bzw. zu klassifizieren und zu strukturieren

genutzt werden. JavaScript ist eine schlanke, dynamisch während der Laufzeit interpretierte Programmiersprache, die speziell für den Einsatz in Web-Browsern konzipiert wurde. Ihr eigentlicher Sprachkern trägt den Namen ECMAScript. Diese Bezeichnung wird jedoch nur selten verwendet. In JavaScript ist es möglich, sowohl prozedural als auch rein funktional bzw. objektorientiert zu programmieren. Mit Javascript lassen sich Aktionen des Nutzers abfangen und Objekte innerhalb der Browser-Umgebung steuern. Beispielsweise kann über das *Document Object Model (DOM)*<sup>5</sup> eine vom Server geladene HTML-Seite direkt im Browser modifiziert werden. Diese hilft Webentwicklern, Teile der Webseite zu animieren, zu entfernen, zu ergänzen bzw. komplett neue Teile dynamisch zu generieren.

## 5.1 Suchinterface

Die Implementierung des Suchinterfaces erfolgte ausschließlich unter Verwendung der eben vorgestellten Web-Technologien; d. h., es kamen keine zusätzlichen Bibliotheken oder Frameworks zum Einsatz. Um eine bessere Übersicht zu gewährleisten und dem Nutzer unnütze Überlegungen zu ersparen, wurden bei der Implementierung nur die Kriterien zur Suche angeboten, die in der Datenbank auch mehrere Werte enthielten. Dazu wurde eine *Structured Query Language (SQL)*-Anfrage realisiert, die jene Parameter zurückgibt, welche mindestens einen Wert besitzen. Diese wurden dann in der Suche dargestellt. Dabei gilt es zwischen Auswahlparameter, für qualitative Werte und Zahlenparameter für quantitative Werte zu unterscheiden. Für die Zahlenwerte wurde eine Berechnung der Minimal- und Maximalwerte für den jeweiligen Parameter vorgenommen, die dann als Anfangswerte bei der größer, kleiner oder im Bereich Suche verwendet wurden. Wählt der Nutzer ein Kriterium aus, das er einschränken möchte und nimmt zum Beispiel die Bereichssuche, wird der Minimal- und Maximalwert für diesen Parameter berechnet sowie bei der Suche in das Eingabefeld standardmäßig eingesetzt, bis er es ändert; d. h., die Bereichssuche würde dann im Bereich von MIN bis MAX heißen, bis der Nutzer einen anderen Bereich angibt. Dies ist notwendig, damit

---

<sup>5</sup>Spezifikation einer Schnittstelle für den Zugriff auf HTML- oder XML-Dokumente

auf der einen Seite ein Kriterium solange unverändert bleibt, bis tatsächlich eine Eingabe eines Wertes vom Nutzer erfolgt und auf der anderen Seite ein Anhaltspunkt für den Wertebereich des Parameters gegeben wird. Die Suchanfrage wird dann mit den eingestellten Kriterien für die Parameter losgeschickt und bezüglich des eingestellten Suchalgorithmus abgearbeitet. Als Ergebnis kommt ein Array mit zwei Tupeln (Index des Aktors und dessen Relevanz) zurück. Dieses Array dient als Grundlage für weitere Abfragen zu den restlichen Daten jedes darin enthaltenen Aktors.

## 5.2 Ergebnisvisualisierung

Für die Entwicklung der Ergebnisvisualisierung musste versucht werden, ein geeignetes Programmier-Framework zu finden, welches die gewünschte Visualisierung zulässt. Die Realisierung einer guten Visualisierung, ohne ein schon vorhandenes Framework zu verwenden, wäre im Rahmen dieser Arbeit nicht zu schaffen gewesen.

Es gibt eine Vielzahl an verfügbaren Diagramm- und Visualisierungs-Frameworks. Zu den Open Source Vertretern zählen die *Google Visualization API*, *Protochart*, *Open Flash Chart 2* und zum Beispiel das *JavaScript Information Visualization Toolkit (JIT)*. Auf der kommerziellen Seite gibt es unter anderen die *Emprise JavaScript Charts*, *PHP/SWF Charts*, *XML/SWF Charts*, *amCharts*, *FusionCharts v3* bzw. *PowerCharts v3*, *Flash Charts Pro* und *anyCharts*.

Diese soeben aufgezählten Visualisierungs-Frameworks wurden hinsichtlich ihres Potentials kurz betrachtet und die vier vielversprechendsten genauer betrachtet. Zu dieser Auswahl zählte das Open Source Framework *Open Flash Chart 2* und die kommerziellen Vertreter *XML/SWF Charts*, *Emprise JavaScript Charts* und *amCharts*. Tabelle 5.1 stellt überblicksartig diese vier Visualisierungs-Frameworks gegenüber. Wie in der Tabelle 5.1 verdeutlicht, bietet das Visualisierungs-Framework *amCharts* die meisten Vorteile. Mit ihm sind alle benötigten Funktionalitäten bzgl. Interaktion realisierbar. Außerdem wird es ständig weiterentwickelt, auch wenn es zu dem Zeitpunkt des Vergleichs schon das aussichtsreichste Framework war. Dies ist ein wichtiger Aspekt, um eine gute Weiterentwicklung der Webanwendung und Zukunftssicherheit hinsichtlich

Tabelle 5.1: Vergleich Visualisierungs-Frameworks

 = Schulnoten	Open Flash Chart 2	XML/SWF Charts	Emprise JavaScript Charts	amCharts
<b>Allgemein</b>				
Lizenz	open source	kommerziell	kommerziell	kommerziell
Preis	0 €	0 - 33 €	0 - 1000 €	0 - 450 €
kostenlose Nutzung	ja	ja, mit kleinen Einschränkungen und ohne Support	ja, mit Einschränkung in Funktionalität & mit Wasserzeichen	ja, mit Firmen Logo (Wasserzeichen)
verglichene Version (Stand: 2009-01-27)	2 Hyperion (Nov 2008)	5.07 (?)	2.0.1 (Sep 2008)	1.6.1.2 (Jan 2009)
aktuelle Version (Stand: 2009-05-13)	2 Ichor (Mär 2009)	5.07 (?)	2.0.1 (Sep 2008)	1.6.3 (Mai 2009)
Aktualität / Updatehäufigkeit	3	4	4	1
<b>Interaktivität</b>				
Zooming	X	X	✓	✓
Scrolling	X	✓	X	✓
Tooltips	✓	✓	✓	✓
Deaktivieren einzelner Datenreihen	X	✓	✓	✓
On Click Events	✓	X	✓	✓
Interaktivität ges.	4	3	2	1
<b>Aussehen</b>				
> 3 verschiedene Formen darstellbar	X	X	X	✓
Verschiedene Farben darstellbar	✓	✓	✓	✓
Anpassbar	✓	✓	✓	✓
Animationen	X	✓	X	✓
Aussehen ges.	3	2	3	1
<b>Entwicklung</b>				
Technologie	Flash	Flash	Javascript	Flash
Plattformunabhängig	✓	✓	✓	✓
PHP Unterstützung	✓	✓	✓	✓
JavaScript Unterstützung	✓	✓	✓	✓
Unterstützte Datenformat	JSON	XML	XML, JSON, CSV, Array	XML, CSV oder dynamisch mit PHP
verfügbare Dokumentation	✓	✓	✓	✓
verfügbarer Sourcecode	✓	X	✓ *1	✓ *1
Entwicklung ges.	2	3	2	2
<b>Gesamt Bewertung</b>	3	3	2	1

\*1 in Entwicklerversion

des Frameworks zu gewährleisten. Der Einstieg in die Umsetzung der geplanten Visualisierung erwies sich dank der guten Dokumentation, der vielen verfügbaren Beispiele und des Online-Forums, als unproblematisch. Mit Hilfe des Online-Forums konnten während der Implementation einige Fragen bezüglich der Umsetzung geklärt und neue Funktionen angefordert werden, die in kommenden Versionen realisiert werden sollen.

## 5.3 Ergebnistabelle

Da es sich beim Design und der Entwicklung der Webanwendung um einen iterativen Prozess zum Erreichen besserer Usability handelt, haben sich manche Designentscheidungen und Funktionalitäten erst während der Umsetzung und der Tests des jeweiligen Prototyps ergeben. Dies führte zu unterschiedlichen Anforderungen an die zu verwendenden JavaScripts und Frameworks. Zum Beispiel wurde die Notwendigkeit der Funktionalität des Filterns der Tabelle und des Verschiebens der Spalten erst in späteren Iterationsschritten aufgedeckt. Um diese Funktionalitäten einfach umzusetzen, stellten sich an die verfügbaren JavaScripts neue Anforderungen. Dies führte teilweise dazu, dass die Wahl des Frameworks entsprechend geändert werden musste. Tabelle 5.2 zeigt eine Übersicht der für die Ergebnistabelle verwendeten Frameworks und JavaScripts. Der Ablauf der Umsetzung der Ergebnistabelle und die damit verbundenen Iterationen können wie folgt beschrieben werden. Eine erste Umsetzung der Ergebnistabelle erfolgte mittels *Table Sorter*. Ein zweiter Prototyp wurde mittels des Frameworks *Google Visualization: Table* realisiert, um der Anforderung hinsichtlich der Balkendarstellung für die einzelnen Werte gerecht zu werden. Dieses Framework zeigte zu dem Zeitpunkt der Implementierung jedoch Einschränkungen bei der Anpassbarkeit des Tabellendesigns und bezüglich der Balkendarstellung.<sup>6</sup> Während einiger Tests zeigte sich außerdem, dass der *Google Visualization: Table* und auch der *Table Sorter* mit dem wissenschaftlichen Zahlenformat (z. B. 2,5e-6) nicht umgehen konnten und bei manchen

---

<sup>6</sup>in Versionen seit 6. April 2009 [GV09] sind jedoch Anpassungen bezüglich Tabellendesign, nicht aber Balkendarstellung, möglich

Tabelle 5.2: Vergleich von Frameworks und JavaScripts für die Ergebnistabelle

<div> <div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> </div> = Schulnoten	Table Sorter	sortable	DataTables	dragtable	Google Visualization: Table	Yahoo! UI Library: DataTable
<b>Allgemein</b>						
Lizenz	open source	open source	open source	?	open source	open source
verglichene Version (Stand: 2009-02-17)	2.0.3 (Mär 2008)	2 (Apr 2007)	1.4.0 (Mär 2009)	1.0 (Jun 2008)	? (Jan 2009)	–
aktuelle Version (Stand: 2009-05-13)	2.0.3 (Mär 2008)	2 (Apr 2007)	1.5 beta 7 (Mai 2009)	1.0 (Jun 2008)	? (Mai 2009)	2.7.0 (2009)
Aktualität / Updatehäufigkeit	4	5	1	3	1	1
<b>Standard Funktionen</b>						
Sortieren	✓	✓	✓	✗	✓	✓
Multi Sort	✓	✗	✓	✗	✓	✓
Korrektes Sortieren	✗	✓	✓ *1	✗	✗	?
Paging	✓	✗	✓	✗	✓	✓
Filter	✗	✗	✓	✗	✗	✓
<b>Erweiterte Funktionen</b>						
Spalten verschieben	✗	✓ *2	✗ *3	✓	✗	✓
Spalten ausblenden	✗	✗	✗	✗	✗	✓
Spaltenbreite verändern	✗	✗	✗	✗	✗	✓
<b>Sonstiges</b>						
Anpassbarkeit	✓	✓	✓	✗	✗ *4	✓
Balkendarstellung	✗	✗	✗	✗	✓	✗
<b>Gesamt Bewertung</b>	3	3	2	5	3	1

\*1 nach Anpassung des JavaScripts

\*2 in Verwendung mit dragtable

\*3 in Verwendung mit dragtable verursacht es Probleme beim Sortieren, kann aber gelöst werden

\*4 in aktueller Version Anpassung möglich



Parametern eine falsche Reihenfolge beim Sortieren produzierten. Ohne eine korrekte Sortierung wäre diese Funktion nutzlos. Dies führte zu der Entscheidung, ein weiteres Framework auszuprobieren und einen Prototypen zu erstellen. Dabei wurde sich eine Möglichkeit überlegt, die Balkendarstellung selbstständig zu programmieren und eine für die Bereichssuche passendere Form zu verwirklichen. Es erfolgte eine Umsetzung der Balken mittels *div*-Tags, während das komplette Design mittels CSS vollständig anpassbar war. Der neu realisierte Prototyp für die Ergebnistabelle wurde mit dem JavaScript *Data Tables* umgesetzt. Hier wurde auch eine Möglichkeit gefunden, ein korrektes Sortieren zu gewährleisten. *Data Tables* ermöglicht außerdem das Filtern, Paging und das Sortieren nach mehreren Spalten (Mutli Sort). Mit Paging ist gemeint, dass die Tabelle nur eine limitierte Anzahl von Ergebnissen darstellt (z. B. 20) und die restlichen auf nachfolgenden Tabellenseiten (Pages) anzeigt.

Die Umsetzung der verschiedenen Interaktionsmöglichkeiten in der Tabelle durch die vorgefertigten JavaScripts war relativ einfach. Auch das Wechseln zwischen den verschiedenen JavaScripts erwies sich als nur wenig aufwendig, da in den meisten Fällen in den *table*-Tag für die Tabellen nur ein spezifischer Klassen Name eingetragen werden musste, damit das JavaScript damit arbeiten konnte. Das zu verwendende JavaScript musste außerdem im Kopfbereich der Webseite eingetragen werden.

Der *Yahoo! UI Library: DataTable* wurde erst nach vollständiger Implementierung des *Data Tables* entdeckt. Dabei handelt es sich um eine komplette UI-Bibliothek und nicht nur um ein Tabellen-Framework. Dies begründet auch seinen so großen Funktionsumfang und die Möglichkeiten der erweiterten Funktionen (Spalten verschieben, ausblenden etc.) im Vergleich zu den anderen Alternativen. Die Verwendung dieser Bibliothek ist vielleicht mit einer schwierigeren Einarbeitung verbunden, zeigt aber auch deutlich mehr Potential bezüglich der Interaktionsmöglichkeiten.

Die alleinige Verwendung des JavaScripts *dragtable* erweist sich auf Grund der vielen fehlenden Funktionen als nicht sinnvoll. Aber in Verbindung mit anderen Frameworks ist sein Einsatz außerordentlich nützlich. *Dragtable* ermöglicht das Verschieben der unterschiedlichen Spalten, um besser zwei Parameter miteinander vergleichen zu können und wichtige Spalten, die ganz hinten in der Tabelle liegen, mehr in den Fokus zu holen.

Für die Umsetzung der Ergebnistabelle wurde ein weiteres JavaScript, die Tooltip-Bibliothek *overLIB*, verwendet. Es gibt viele Tooltip-Bibliotheken, jedoch musste eine gefunden werden, die erstens in Verbindung mit der Verwendung der anderen JavaScripts keine Probleme verursacht und zweitens die Möglichkeit anbietet, einen eigenen Hintergrund für den Tooltip zu verwenden. Das Erstellen und Verwenden des eigenen Hintergrundes war notwendig, um mit dem in der Ergebnisvisualisierung verwendeten Tooltip-Design übereinzustimmen und somit ein konsistentes Design zu gewährleisten. Während der Implementation stellte sich heraus, dass die gleichzeitige Verwendung von JavaScripts, welche auf dem Framework *MooTools* und *jQuery* basieren, zu Problemen führte. Deswegen wurde bei der Umsetzung darauf geachtet, dass die verwendeten JavaScripts ausschließlich auf dem Framework *jQuery* basieren.

# Kapitel 6

## Zusammenfassung und Ausblick

### 6.1 Zusammenfassung

Bei der Entwicklung mechatronischer Systeme besteht die Schwierigkeit, für eine gegebene Problemstellung den optimalen Antrieb zu wählen. Dabei muss eine Vielzahl von Kriterien und Eigenschaften beachtet werden. Bisher wurden für die Auswahl von Aktoren stets klassische Konstruktions- und Produktkataloge herangezogen (siehe [Erb09]). Der Auswahlprozess war aber immer sehr zeitintensiv und oft nicht herstellerübergreifend. Ziel dieser Arbeit ist die Entwicklung und Implementation einer Webanwendung für die herstellerübergreifende Suche von Direktantrieben und deren Ergebnisvisualisierung, mit dem Schwerpunkt auf Usability und UI-Design.

Um eine benutzerfreundliche, effektive und effiziente Aktorsuchen zu gewährleisten, wurde nach dem UEL von [May99] vorgegangen. Dabei wurde mit der Anforderungsanalyse, die u. a. die Erstellung der Nutzerprofile, die Aufgabenanalyse und das Ableiten von Usability-Zielen vorsah, begonnen.

Danach wurde das konzeptionelle Design und darauf aufbauend das Screen und detaillierte UI-Design entwickelt. In dieser Phase wurden verschiedene Visualisierungsmöglichkeiten miteinander verglichen und letztendlich eine Form des Scatter Plots, der Bubble Chart, für die Ergebnisdarstellung umgesetzt. Eine erweiterte Tabelle mit integrierter Balkendarstellung kam für das Anzeigen der Details der Aktoren zum Einsatz.

Umgesetzt wurde die Ergebnisdarstellung mit Hilfe des Visualisierungs-Frameworks *amCharts* (für den Bubble Chart) und dem für das JavaScript Framework *jQuery* verwendeten Plug-In *Data Tables* (für die Tabellendarstellung). Bei dem Design, dem Erstellen der Prototypen und den Tests (Evaluation) handelt es sich um einen iterativen Prozess, der dazu dienen sollte die vorhandenen Usability-Probleme aufzudecken. Am Ende entstand eine voll funktionsfähige Webanwendung, welche die gezielte Suche und die freie Exploration von Direktantrieben ermöglicht.

## 6.2 Ausblick

Auch wenn die Webanwendung voll funktionsfähig ist, ergeben sich doch ständig Verbesserungen, die es umzusetzen gilt. Dies ist in der Software- und auch in der Webentwicklung gang und gäbe. Der folgende Abschnitt beschreibt einen Ausblick hinsichtlich weiterer Möglichkeiten in den Bereichen: Suchinterface, Ergebnisvisualisierung, Ergebnistabelle und sonstigen Funktionen. Diese Möglichkeiten könnten zum Beispiel im Rahmen einer Studententätigkeit ergänzt werden. Da der Umfang dieser Arbeit keine umfassenden Usability-Tests erlaubte, müssen diese ebenso in einer weiterführenden Arbeit durchgeführt werden.

### 6.2.1 Evaluation

Für eine Evaluation der Webanwendung würden sich zwei Möglichkeiten bieten. Die erste sind umfangreiche Benutzer-Tests (Usability-Tests) in einem Usability-Labor. Dabei würde dem Nutzer eine Aufgabe (ein typischer Anwendungsfall) gestellt werden, die er zu lösen hat. Bei dem Lösen der Aufgabe soll der Nutzer seine Gedankengänge laut äußern. Dabei wird er von zwei Kameras aufgenommen. Während eine die Aktionen und Navigationswege, die der Nutzer am Bildschirm durchführt, aufnimmt, schneidet die andere Kamera die Reaktionen, Gestik und Mimik des Nutzers mit. Zusätzlich nimmt ein Mikrofon die gesprochenen Gedanken auf, die der Nutzer während seiner Tätigkeit von sich geben soll. Ziel des Usability-Tests ist es, Stärken und Schwächen bzgl. der Usability-Kriterien zu erkennen und Usability-Probleme aufzudecken.

Ein Usability-Test besteht aus den folgenden vier Schritten:

1. Vorbereitung,
2. Einführung,
3. Durchführung des Tests und
4. Auswertung.

Die zweite Möglichkeit einer Evaluation besteht darin, die erstellte Webanwendung über das Internet durch einen ausgewählten Nutzerkreis für Testzwecke zugänglich zu machen. Die Nutzer bekommen eine Aufgabe zugeteilt, die einen typischen Anwendungsfall vorsieht. Nach der Bearbeitung dieser Aufgabe bekommen sie einen Online-Fragebogen, um die Benutzung und Aspekte der Webanwendung zu bewerten. Dabei könnte auch ein Vorher-/Nachher-Test genutzt werden, um vergleichende Aussagen über Effektivität, Effizienz und Zufriedenheit beider Varianten vornehmen zu können. Bei diesem Vorher-/Nachher-Test könnte die Listendarstellung, wie man sie von herkömmlichen Suchsystemen kennt, mit der neuen Bubble Chart Visualisierung und zugehöriger Ergebnistabelle mit integrierter Balkendarstellung verglichen werden. Dafür wäre es jedoch notwendig, die entwickelte Ergebnisvisualisierung und die Ergebnistabelle zu entfernen und die Ergebnisse lediglich durch eine Liste oder evtl. auch durch eine normale Tabelle darstellen zu lassen. Ein weiterer Vorher-/Nachher-Test könnte zusätzlich durchgeführt werden, um die zwei verschiedenen Varianten des Suchinterfaces (siehe Kapitel 4.3.1) zu vergleichen.

Eine Möglichkeit, eine Online-Befragung zu realisieren, bietet *AttrakDiff* [HBK03] mit ihrem *AttrakDiff2* Fragebogen auf [www.attrakdiff.de](http://www.attrakdiff.de) an. Damit lassen sich Webanwendungen beurteilen und es lässt sich ermitteln, wie attraktiv, im Hinblick auf Bedienbarkeit und Aussehen die getestete Webanwendung empfunden wird und ob Optimierungsbedarf besteht. Die aus der Evaluation erlangten Ergebnisse können direkt für das Redesign der Webanwendung verwendet werden.

### 6.2.2 Suchinterface

Die in Kapitel 4.3.1 vorgestellten zweiten Varianten des Suchinterfaces (siehe Abbildung 4.3 und 4.4) könnten umgesetzt und durch einen Usability-Test mit den Ergebnissen der ersten Variante (Abbildung 4.2) verglichen werden. In Abhängigkeit der Ergebnisse dieses Tests könnte über eine endgültige Realisierung dieser zweiten Variante nachgedacht werden. Die zweite Variante würde es ermöglichen, dass nur die Kriterien dargestellt werden, nach denen auch gesucht wurde. Dies würde zu mehr Übersichtlichkeit führen. Dabei könnte die Eingabe für die quantitativen Parameter zusätzlich über einen Schieberegler (Slider), ähnlich wie beim *FilmFinder*, realisiert werden (siehe Abbildung 4.4). Des Weiteren könnte über eine Möglichkeit nachgedacht werden, wie zwei Parameter miteinander verknüpft werden könnten, um somit das Produkt oder die Differenz beider Parameter zu berechnen. Dieser berechnete Wert müsste dann in der Visualisierung als Größe für eine anzuzeigende Dimension (X-/Y-Achse oder Größe der Blase) auswählbar sein. Bei dieser Idee wäre es auch sinnvoll, den berechneten Wert ebenso bei der Ergebnisrelevanz mit einfließen zu lassen.

### 6.2.3 Ergebnisvisualisierung

Weitere Ideen und Funktionen, die die Visualisierung betreffen, sind folgende. Um dem Nutzer mehr Möglichkeiten zum Anpassen der in der Visualisierung dargestellten Parameter zu geben, könnten, wie bei der Wahl für die Größe der Blasen, zusätzliche Auswahlboxen für die Wahl der Farbe und evtl. auch für die Wahl der Form genutzt werden. Bisher gibt die Farbe nur die verschiedenen Aktortypen wieder, wobei auch eine Verwendung für die verschiedenen Hersteller sinnvoll wäre. Da *amCharts* neben den Kreisen (Blasen) auch noch andere Formen unterstützt, müsste über eine adäquate Verwendung nachgedacht werden. Es könnten sich damit andere qualitative Parameter darstellen lassen. Im Allgemeinen muss aber darauf geachtet werden, dass die kognitiven Fähigkeiten des Nutzers nicht zu sehr belastet werden und die Visualisierung dadurch zu unübersichtlich und unverständlich wird.

Bisher ist eine Einschränkung der Suche nur mit Hilfe des Suchinterfaces möglich. Die Visualisierung ermöglicht zwar einen Bereich zu vergrößern, jedoch nicht die Suche für

diesen Bereich auch weiter einzuschränken. Eine Möglichkeit, eine neue Suchanfrage über den rangezoomten Bereich auszuführen, wäre wünschenswert. Dabei könnte diese Funktion durch gemeinsames Zoomen und Drücken der zum Beispiel Alt-Taste erreicht werden. Das normale Zoomen würde davon unberührt bleiben. Dies hat den Vorteil, dass dieses Feature nachträglich eingebaut werden könnte und sich der Nutzer nicht umgewöhnen müsste.

Bislang besteht auch keine Möglichkeit, bei einem mehrfach rangezoomten Bereich, zu der vorhergehenden Zoomstufe zurückzuspringen. Es müsste über eine für den Nutzer leicht verständliche Art, diese Option anzubieten, z. B. im Rechtsklick-Menü der Maus, nachgedacht werden.

Außerdem könnten die Aktoren in der Visualisierung anklickbar gemacht werden, um sie dann in der Tabelle hervorheben zu können. Durch die gute JavaScript-Unterstützung von *amCharts* müsste dies leicht realisierbar sein. Weitere Informationen dazu finden sich in der Online-Dokumentation von *amCharts* (siehe [amC09]). Auch wenn sich der Scatter Plot und der Bubble Chart als beste Möglichkeit zur Darstellung der Suchergebnisse hypervarianter Daten gezeigt hat, gibt es doch auch Fälle (wenig Suchergebnisse), in denen eine Visualisierung als Bar Chart oder als Star Plot sinnvoll wäre. Evtl. ist es nützlich, in diesen Fällen dem Nutzer die Wahl zwischen verschiedenen Visualisierungsmöglichkeiten anzubieten. Mit Hilfe von Tabs könnte er zwischen den verschiedenen Darstellungsformen aussuchen und eine wählen, welche gerade für seine Bedürfnisse am besten geeignet ist.

#### 6.2.4 Ergebnistabelle

Auch in den Interaktionsmöglichkeiten bzgl. der Tabelle steckt weiteres Verbesserungspotential. Bisher ist zwar das Verschieben von einzelnen Spalten möglich, verursacht aber bei vertauschter Spaltenreihenfolge Probleme beim Sortieren. Es müsste also eine funktionsfähige Möglichkeit zum Vertauschen der Spalten angeboten werden. Ebenso wäre es sinnvoll, eine Konfigurationsmöglichkeit für die Tabelle anzubieten, damit der Nutzer die Möglichkeit hat, Spalten, die ihm nicht interessieren, ausblenden zu können.

### 6.2.5 Sonstige Funktionen

Weiteres Verbesserungspotential der Webanwendung zeigt sich hinsichtlich der Möglichkeit, sowohl die gestellte Suchanfrage, als auch die Aktorsuchergebnisse abspeichern zu können.

Auch die Möglichkeit Notizen zu einzelnen Ergebnissen (Aktoren) machen zu können und diese in der Datenbank zu speichern, könnte sich als nützlich erweisen. Der *Yahoo! UI Library: DataTable* [Yah09] bietet diese Möglichkeit und könnte in einer zukünftigen Version der Webanwendung dafür genutzt werden.

Eine weitere Verbesserungsmöglichkeit besteht in einer optimierten Druckausgabe der Aktorsuche. Diese sollte in jedem Fall die Suchanfrage mit den eingestellten Kriterien und die Ergebnistabelle mit den Details einschließen. Optional kann auch die Visualisierung (für die Übersicht der Aktorsuchergebnisse) beim Drucken ergänzt werden. Um den Nutzern mehr Möglichkeiten zu geben und gerade erfahrene Nutzer besser zu unterstützen, sollten auch allgemeine Konfigurationsmöglichkeiten (z. B. bezüglich des Tabellenlayouts und der Visualisierungsparameter) angeboten werden.



# Literaturverzeichnis

- [amC09] AMCHARTS: *amCharts Online Dokumentation*. <http://www.amcharts.com/docs/v.1/bundle>. Version: 2009, Abruf: Mai 2009
- [AS94] AHLBERG, Christopher ; SHNEIDERMAN, Ben: Visual information seeking using the FilmFinder. In: *CHI Conference Companion, Conference on Human Factors in Computing Systems*. Boston, Massachusetts, USA : ACM, April 1994, 433–434
- [BKS01] BÖRZSÖNYI, Stephan ; KOSSMANN, Donald ; STOCKER, Konrad: The Sky-line Operator. In: *Proceedings of the 17th International Conference on Data Engineering*. Washington, DC, USA : IEEE Computer Society, 2001. – ISBN 0-7695-1001-9, S. 421–430
- [Bra06] BRANDT, Matthias: *Entwurf und prototypische Realisierung einer multimedialen Mechanismen- und Getriebebibliothek*, Technische Universität Ilmenau, Diplomarbeit, 2006
- [Dem07] DEMARMELS, Mischa: *Konzeption und Entwicklung einer Visualisierung für mehrdimensionale Daten*, Universität Konstanz Konstanz, Diplomarbeit, 2007
- [DIN03] *Software-Ergonomie - Empfehlungen für die Programmierung und Auswahl von Software*. Beuth, 2003
- [Erb09] ERBE, Torsten: *Grundlagen und Ansätze zur Entscheidungsunterstützung bei der Antriebsauswahl*, TU Ilmenau, Diplomarbeit, 2009

- 
- [GV09] *Google Visualization API: Release Notes*. [http://code.google.com/apis/visualization/documentation/release\\_notes.html](http://code.google.com/apis/visualization/documentation/release_notes.html). Version: Mai 2009
- [Han76] HANSEN, Friedrich: *Konstruktionswissenschaft*. VEB Verlag Technik Berlin, 1976 (2. Auflage)
- [HBK03] HASSENZAHL, Marc ; BURMESTER, Michael ; KOLLER, Franz: AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität. 2003. – Forschungsbericht
- [Hel06] HELLBUSCH, Jan E.: *Barrierefreies Webdesign*. Bd. 1. Auflage. dpunkt Verlag, 2006
- [HV03] HEINSEN, Sven ; VOGT, Petra: *Usability praktisch umsetzen*. Hanser Fachbuchverlag, 2003
- [KB91] KALLENBACH, E. ; BÖGELSACK, G.: *Gerätetechnische Antriebe*. Bd. 1. Auflage. Berlin: Verlag Technik, 1991
- [Lud04] LUDWIG, Kerstin A.: *STAR: Visualisierung von Daten*. April 2004
- [May99] MAYHEW, Deborah J.: *The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design*. 1. Auflage. Morgan Kaufmann, 1999
- [Möh06] MÖHRKE, Carsten: *Besser PHP programmieren : professionelle PHP-Techniken*. 2. Auflage. Bonn: Galileo-Press, 2006
- [Naj03] NAJARRO, Martha Argentina B.: *Visualisierung von Informationsräumen*, Technische Universität Ilmenau, Diss., 2003
- [Nie93] NIELSEN, Jakob: *Usability Engineering*. 1. Auflage. Morgan Kaufmann, 1993
- [RC01] ROSSON, Mary B. ; CARROLL, John M.: *Usability Engineering: Scenario-Based Development of Human Computer Interaction*. 1. Auflage. Morgan Kaufmann, 2001

- 
- [Rei05] REITERER, Harald: Visuelle Exploration Digitaler Datenbestände. In: EIBL, Maximilian (Hrsg.) ; REITERER, Harald (Hrsg.) ; STEPHAN, Peter F. (Hrsg.) ; THISSEN, Frank (Hrsg.): *Knowledge Media Design. Theorie, Methodik, Praxis*. Oldenbourg Wissenschaftsverlag GmbH, 2005
- [RF07] RICHTER, Michael ; FLÜCKIGER, Markus: *Usability Engineering kompakt: Benutzbare Software gezielt entwickeln*. Spektrum akademischer Verlag, 2007
- [Shn94] SHNEIDERMAN, Ben: Dynamic queries for visual information seeking. In: *Software, IEEE* 11 (1994), Nov, Nr. 6, S. 70–77
- [Shn02] SHNEIDERMAN, Ben: *User Interface Design*. 3. Auflage. mitp-Verlag, Bonn, 2002
- [SM00] SCHUHMANN, H. ; MÜLLER, W.: *Visualisierung, Grundlage und allgemeine Methoden*. Berlin: Springer Verlag, 2000
- [SP04] SHNEIDERMAN, Ben ; PLAISANT, Catherine: *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 4. Auflage. Pearson Education, 2004
- [Spe07] SPENCE, Robert: *Information Visualization: Design for Interaction*. 2. Auflage. Prentice Hall, 2007
- [SVM99] SEBRECHTS, M. ; VASILAKIS, J. ; MILLER, M.: *Visualization of Search Results: A Comparative Evaluation of Text, 2D, and 3D Interfaces*. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval Conference (pp. 3-10)*. New York: ACM Press, 1999
- [Tuf83] TUFTE, Edward R.: *The visual display of quantitative information*. Graphics Press, 1983
- [Ver09] *Wachstums Kern VERDIAN - VERnetzte Integrierte Magnetische DIREKTANtriebe*. <http://www.verdian.eu>. Version: 2009, Abruf: Mai 2009

- 
- [w3s08] W3SCHOOLS.COM: *Browser Statistik*. [http://www.w3schools.com/browsers/browsers\\_stats.asp](http://www.w3schools.com/browsers/browsers_stats.asp). Version: Oktober 2008
- [w3s09] W3SCHOOLS.COM: *Browser Display Statistik*. [http://www.w3schools.com/browsers/browsers\\_display.asp](http://www.w3schools.com/browsers/browsers_display.asp). Version: Januar 2009
- [Win08] WINKLER, Mike: *Konzeption und Umsetzung eines Bewertungs- und Auswahltools für Antriebssysteme*, TU Ilmenau, Diplomarbeit, 2008
- [Yah09] *Yahoo! UI Library: DataTable*. <http://developer.yahoo.com/yui/datatable>. Version: 2009, Abruf: Mai 2009
- [Zan07] ZANDONÁ, Guilherme L.: *Entwicklung eines Konzepts zur Bewertung und Auswahl von Antriebssystemen in frühen Entwurfsphasen*, TU Ilmenau, Diplomarbeit, 2007

## Anhang A

### Nutzerprofil-Fragebogen

# Nutzerprofil

## Nutzerkategorie Identifikatoren

### 1. In welchem Bereich sind Sie tätig?

☐ Universität. ☐ Industrie.

### 2. Zu welchem Fachbereich gehören Sie?

☐ Elektrotechnik. ☐ Maschinenbau. ☐ Mechatronik.

### 3. Über welche Ausbildung verfügen Sie?

☐ Abitur (Student). ☐ Abgeschlossenes Hochschulstudium.

## Physische Eigenschaften

### 4. Welches Geschlecht sind Sie?

☐ Männlich. ☐ Weiblich.

### 5. Wie alt sind Sie?

☐ 18-28. ☐ 29-40. ☐ 41-55. ☐ über 55.

### 6. Sind Sie Farbenblind?

☐ Ja. ☐ Nein. ☐ Weiß nicht.

### 7. Tragen Sie eine Brille oder Kontaktlinsen?

☐ Nein. ☐ Ja.

### 8. Haben Sie eine physische Beeinträchtigung?

☐ Nein. ☐ Ja.

### Einstellung und Motivation

#### 9. Wie ist Ihre generelle Einstellung zu Computern?

- ☐ Abgeneigt (Ich **mag es nicht** mit Computern zu arbeiten).  
☐ Neutral.  
☐ Aufgeschlossen (Ich **mag es** mit Computern zu arbeiten).

#### 10. Ist die Zeit, um sich in eine neue Software einzuarbeiten, es normaler Weise wert?

- ☐ Ja. ☐ Manchmal. ☐ Nein.

#### 11. Macht es Ihnen Spaß das Bedienen neuer Software zu erlernen?

- ☐ Ja. ☐ Manchmal. ☐ Nein.

#### 12. Wie ist Ihre generelle Einstellung zum Internet?

- ☐ Abgeneigt (Ich **mag es nicht** im Internet zu surfen).  
☐ Neutral.  
☐ Aufgeschlossen (Ich **mag es** im Internet rumzusurfen)

### Wissen und Erfahrungen

#### 13. Was ist Ihre Muttersprache?

- ☐ Deutsch (weiter mit Frage 15). ☐ Englisch. ☐ andere.

#### 14. Wenn Ihre Muttersprache nicht deutsch ist, wie gut verstehen Sie dann deutsch?

- ☐ Schlecht. ☐ Mittel. ☐ Gut.

#### 15. Wie würden Sie Ihre Computerkenntnisse beschreiben?

- ☐ Keine Erfahrung.  
☐ Wenig.  
☐ Mittel.  
☐ Gut.

**16. Wie würden Sie Ihre Kenntnisse im Umgang mit dem Internet beschreiben?**

\_\_\_\_\_ Keine Erfahrung.    \_\_\_\_\_ Wenig.    \_\_\_\_\_ Mittel.    \_\_\_\_\_ Gut.

**17. Wie viel Zeit verbringen Sie wöchentlich am Computer?**

\_\_\_\_\_ Wenig (< 5h).    \_\_\_\_\_ Mittel (5-15h).    \_\_\_\_\_ Viel (>15h).

**18. Welche Aufgaben erledigen Sie mit Ihrem Computer und wie viel Zeit setzen Sie dafür wöchentlich ein (geschätzt)?**

Anwendungsbereich	Dauer (in Stunden)
1. Emails und andere Kommunikation	_____
2. Textverarbeitung	_____
3. Recherche	_____
4. Internet (ohne Recherche)	_____
5. _____	_____
6. _____	_____

**19. Nennen Sie alle Programme die Sie aktuell in Ihren Job nutzen, und geben Sie an, wie lange Sie sie schon benutzt haben.**

Programm / Software	Erfahrung + Dauer (Jahre)
1. MS Word	_____
2. MS Excel	_____
3. Outlook	_____
4. _____	_____
5. _____	_____
6. _____	_____

**Job und Umgebung****20. Ungefähr wie viel Prozent Ihrer Arbeitszeit verbringst du an folgenden Orten?**

\_\_\_\_\_ Büro.    \_\_\_\_\_ Werkstatt.    \_\_\_\_\_ Anderer Ort.



**21. Wie lange sind Leute im Durchschnitt mit dem Projekt beschäftigt?**

\_\_\_\_\_ < 6 Monate.      \_\_\_\_\_ 6 - 12 Monate.      \_\_\_\_\_ 1 - 3 Jahre.  
\_\_\_\_\_ Keine Ahnung.

**22. Über welche Ausstattung verfügen Sie an Ihrer Arbeit?**

Ausstattung	Eigenschaft
1. Grösse des Monitors (in Zoll)	_____
2. Auflösung mit der gearbeitet wird	_____
3. Internetanbindung (z.B. in Kbit/sec)	_____
4. Prozessor (falls bekannt)	_____
5. Arbeitsspeicher RAM (in GB)	_____
6. Betriebssystem (Win 98, 2000, XP, Vista)	_____
7. Internet Browser (IE 6, IE 7, Firefox, ...)	_____

**Aufgabenbereiche****23. Wie häufig suchen Sie nach Antrieben bzw. Antriebssystemen und wie viel Zeit verwenden Sie im Durchschnitt dafür?**

Häufigkeit (wie oft: z.B. 2x im Monat): \_\_\_\_\_ .

Dauer pro Suche (wie viel Stunden): \_\_\_\_\_ .

**24. Welche Methoden haben Sie bisher für die Suche (Recherche) nach Antrieben bzw. Antriebssystemen genutzt? (kurz beschreiben welche Form von Suchdiensten, Portalen, Datenbanken o.ä. benutzt wurden)**

---

---

---

---

## Anhang B

### Aufgabenanalyse-Fragebogen

# Aufgabenanalyse

1. Welche Methoden haben Sie bisher für die Suche (Auswahl) von Aktoren genutzt? (kurz beschreiben wie Sie momentan bei der Suche nach passenden Antrieben für eine gegebene Problemstellung vorgehen, z. B. welche Form von Suchdiensten, Portalen, Datenbanken o. ä. benutzt werden)

---

---

---

2. Wie häufig suchen Sie nach Aktoren und wie viel Zeit verwenden Sie im Durchschnitt dafür (schätzungsweise)?

Häufigkeit (wie oft: z. B. 2x im Monat): \_\_\_\_\_ .

Dauer pro Suche (wie viel Stunden/Minuten): \_\_\_\_\_ .

3. Haben Sie schon mal die Erweiterte Suche bzw. Experten Suche bei der Suche im Internet verwendet?

\_\_\_\_\_ Ja. \_\_\_\_\_ Nein.

4. Mit welchem Browser arbeiten sie? (auch mehrfach Nennung möglich)

\_\_\_\_\_ Internet Explorer 7. \_\_\_\_\_ Firefox. \_\_\_\_\_ Google Chrome.

\_\_\_\_\_ Internet Explorer 6. \_\_\_\_\_ Safari. \_\_\_\_\_ Opera.

\_\_\_\_\_ Anderer: \_\_\_\_\_ .

5. Unter welcher Zeitvorgabe sollte die zukünftige Suche erledigt werden?

---

---

---

- 6. Wie viele und welche Größen sind bei der Suche nach Aktoren im Durchschnitt relevant?**

---

---

---

- 7. Wie sollte aus Ihrer Sicht solch ein Such-Interface aussehen (beschreiben und/oder grob skizzieren)?**

---

---

---

---

---

---

- 8. Wie sollte aus Ihrer Sicht die Ergebnisdarstellung (Visualisierung der Ergebnisse) für diese Suche aussehen (beschreiben und/oder grob skizzieren)?**

---

---

---

---

---

---

- 9. Wenn keine Ergebnisse oder zu viele Ergebnisse bei der Suche gefunden werden, welches Feedback und weiteren Aktionen würden Sie sich vom Such-System wünschen (wie könnte der Nutzer bei der Auswahl unterstützt werden)?**

---

---

---

---

---

---

**10. Gibt es sonstige Funktionen, die Sie sich von einem System zur Suche und Auswahl von Akteuren wünschen würden?**

---

---

---

---

---

---

**11. Versuchen Sie mit ca. 4 Eigenschaften Ihre Vorstellungen und Ansprüche gegenüber einem solchen System zu beschreiben.**

---

---

---

---

# Eidesstattliche Erklärung

Ich versichere, dass ich diese Diplomarbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Ilmenau, 27.05.2009

Torsten Meß

# Thesen

1. Bei den Nutzern, welche die Webanwendung zur Aktorauswahl verwenden, handelt es sich um eine homogene Nutzergruppe.
2. Die Eingrenzung der Aktoren bei der Suche sollte trichterförmig geschehen.
3. Für die Aktorauswahl muss sowohl die gezielte Exploration (Suche) als auch die freie Exploration (Browsen) unterstützt werden.
4. Der Scatter Plot und Bubble Chart eignen sich am besten für die intuitive und übersichtliche Visualisierung von Aktorsuchergebnissen.
5. Die Tabelle eignet sich am besten für die Darstellung von Details der multidimensionalen Daten.
6. Die Balkendarstellung in der Ergebnistabelle erhöht die Übersichtlichkeit.

Ilmenau, 27.05.2009

Torsten Meß